

UNIVERSIDADE FEDERAL DO PARANÁ - SETOR LITORAL

THIAGO BERNARDI CALEGARI

BANCO DE DADOS PARA UM SISTEMA WEB ORIENTADO A SERVIÇOS
RELACIONADO À CONSTRUÇÃO CIVIL NO LITORAL PARANAENSE

MATINHOS

2015

THIAGO BERNARDI CALEGARI

BANCO DE DADOS PARA UM SISTEMA WEB ORIENTADO A SERVIÇOS
RELACIONADO À CONSTRUÇÃO CIVIL NO LITORAL PARANAENSE

Trabalho apresentado como requisito parcial à obtenção do grau de Bacharel em Informática e Cidadania no Curso de Graduação em Informática e Cidadania da Universidade Federal do Paraná – Setor Litoral.


Orientador: Prof. Me. Paulo Gaspar Graziola Junior.

MATINHOS

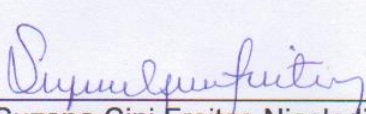
2015

ATA DE AVALIAÇÃO DA DEFESA DO TRABALHO DE CONCLUSÃO DE CURSO


Aos dois dias do mês de julho de dois mil e quinze, às 20 horas, no Setor Litoral da Universidade Federal do Paraná, reuniu-se a banca avaliadora do trabalho de conclusão de curso, constituída pelo professor Me. Marcio Hosoya Name e pela professora Dra. Suzana Cini Freitas Nicolodi sob a presidência do Orientador, Professor Me. Paulo Gaspar Graziola Junior. O Trabalho de Conclusão do Curso de Bacharelado em Informática e Cidadania, do aluno Thiago Bernardi Calegari, sob o título: "BANCO DE DADOS PARA SISTEMA WEB ORIENTADO A SERVIÇOS RELACIONADOS A CONSTRUÇÃO CIVIL NO LITORAL PARANAENSE", obteve o conceito APL. O aluno deverá efetuar as correções solicitadas pela banca e entregar a versão final em formato digital via CD-ROOM, até o dia 10 de junho de dois mil e quinze, na assessoria a Câmara do curso de Informática e Cidadania.



Paulo Gaspar Graziola Junior
Professora Orientadora

Paulo Gaspar Graziola Junior
Docente
Matr. 203045
Setor Litoral – UFPR


Suzana Cini Freitas Nicolodi
Membro da banca avaliadora

Suzana Cini Freitas Nicolodi
Docente
Matrícula 202246
Setor Litoral – UFPR


Marcio Hosoya Name
Membro da banca avaliadora


Thiago Bernardi Calegari

Prof. Me. Paulo Gaspar Graziola
Junior.
Professor Orientador

Prof. Márcio Hosoya Name.

Prof^a. Suzana Cini Freitas Nicolodi.

AGRADECIMENTOS

À minha amada Mãe, Lenir de Fátima Bernardi, por tudo, desde a concepção até o dia de hoje, todo o carinho nos momentos mais difíceis da minha vida e, principalmente, por todas as vezes em que ela me disse “Não se esqueça que eu te amo”.

Ao meu Pai, Amarildo Batista Calegari, pelo apoio, incentivo e ajuda que prestou durante esta jornada.

Aos professores da UFPR Litoral, de maneira particular, aos professores Paulo Gaspar Graziola Junior e Neilor Firmino Camargo, ambos formados na área de informática, pelos conselhos, direcionamentos e orientações que prestaram, tanto em sala de aula, como nos corredores da universidade.

Aos professores que fazem parte da minha banca, Márcio Hosoya Name e Suzana Cini Freitas Nicolodi.

Aos meus colegas de sala de aula Adenilson de Paiva Domingues, Gilson Pereira dos Santos, Luciano Oliveira dos Santos, Emerson de Lima e Erick Penkal Matoso, pela oportunidade de eu participar da concepção do projeto Construtor Civil, futura ferramenta de pesquisa *Web* voltado para a construção civil no litoral do Paraná, que aborda, literalmente, Informática e Cidadania.

Finalmente, um colega meu, aliás, um grande amigo, Victor Hugo da Silva, por toda a força e todo o incentivo que me deu desde a inscrição para o vestibular na UFPR Litoral até os dias de hoje, pois ele nunca me deixou desistir das oportunidades.

“Muita gente, ao errar na
avaliação dos outros, achará
justo fazer o mal aos amigos
por julgar que sejam maus.
Mas favorecerá aos inimigos,
pensando que sejam bons”.

Sócrates

RESUMO

O objetivo deste trabalho é descrever o desenvolvimento de um sistema Web utilizando o *framework* “PHPMyAdmin” com o intuito de auxiliar nos processos de cadastro de profissionais, produtos e serviços no ramo da construção civil, tendo como cenário os sete municípios do litoral paranaense. Será discutida a base de dados, sua concepção, suas características e seus benefícios em relação a forma de gestão utilizada pelo portal eletrônico Construtor Civil. O conhecimento empírico de clientes e profissionais do ramo serviu de ferramenta no fomento de ideias, facilitando o processo de construção do sistema. Também serão apresentadas as ferramentas utilizadas na montagem do sistema de base de dados, dando preferência nas escolhas para tecnologias orientadas a páginas Web, salientando a necessidade de uma política que proporcione maior número de possibilidades de interação entre o cliente e o sistema, contribuindo para a agilidade nos processos de cadastro, atualização e consulta de produtos, fornecedores, serviços, produtos e prestadores de serviços, reduzindo a peregrinação pelas cidades do litoral do Paraná, em busca de produtos e serviços relacionados à construção civil na região. A metodologia utilizada se dá por meio de levantamento de requisitos, estudos de caso de uso, testes do banco e criação de diversas versões até a que apresentar maturidade suficiente para que seja declarada como a versão final. Um mapa conceitual foi necessário para adquirir entendimento do plano de negócio proposto para o portal eletrônico, o qual receberá o banco de dados. O resultado é uma base de dados coerente com a proposta dos desenvolvedores do portal eletrônico Construtor Civil, além da rapidez e organização estruturada que o banco proporciona para o sistema *WEB*. E ainda, pensa-se em migrar o banco de dados para uma nova modalidade em desenvolvimento de sistemas de base de dados, conhecida como banco de dados não-relacional NoSQL, o qual acredita-se propiciar uma maior rapidez na busca de informações.

Palavras-chaves: Sistema *WEB*, Banco de dados, desenvolvimento de sistemas, construção civil.

ABSTRACT

The objective of this study is to describe the development of a Web system using the framework "PHPMyAdmin" which aims to assist in the professional registration processes, products and services in the construction sector, against the backdrop of the seven municipalities of the Paraná coast. It will discuss the database, its design, its characteristics and its benefits regarding the management approach used by the Civil Constructor electronic portal. The empirical knowledge of customers and industry professionals served as a tool in the promotion of ideas, facilitating system construction process. Also will be presented the tools used in assembling the database system, giving preference to the choices oriented Web pages technologies, stressing the need for a policy that provides more possibilities of interaction between the client and the system, contributing to the agility registration processes, update and product consultation, suppliers, services, products and service providers, reducing the pilgrimage through the cities of the coast of Paraná, in search of products and services related to construction in the region. The methodology used is through lifting requirements, use case studies, bench testing and creating several versions until the one with sufficient maturity to be declared as the final version. A conceptual map was needed to acquire understanding of the business plan proposed for the electronic portal, which will receive the database. The result is a consistent database with the proposal of the developers of electronic Civil Constructor portal, as well as quickly and structured organization that the bank provides to the WEB system. And yet, it is thought to migrate the database to a new modality in developing database systems, known as non-relational database NoSQL, which is believed to provide greater speed in finding information.

Keywords: Web System, Database, system development, civil construction.

LISTA DE FIGURAS

FIGURA 1: TELA INICIAL DO LINUX – DISTRO UBUNTU VERSÃO 14.04 LTS 64 BITS.....	28
FIGURA 2: TELA INICIAL DO WINDOWS 8.....	30
FIGURA 3: DEMONSTRAÇÃO DE FUNCIONAMENTO DO MYSQL CLIENT.....	32
FIGURA 4: MICROSOFT VISIO RODANDO NO SISTEMA OPERACIONAL WINDOWS 8.....	35
FIGURA 5: MOZILLA FIREFOX.....	37
FIGURA 6: PHPMYADMIN.	39
FIGURA 7: MYSQL WORKBENCH.	40
FIGURA 8: MAPA CONCEITUAL DO SISTEMA.	43
FIGURA 9: MODELO ENTIDADE-RELACIONAMENTO ENTRE OS PADRÕES DE CONSTRUÇÃO E AS PLANTAS E MODELOS ARQUITETÔNICOS.....	44
FIGURA 10: MODELO ENTIDADE-RELACIONAMENTO ENTRE A CIDADE E OS CLIENTES E OS FORNECEDORES DE MATERIAIS.	45
FIGURA 11: MODELO ENTIDADE-RELACIONAMENTO – SUBSISTEMA AVALIAÇÃO E REPUTAÇÃO.....	47
FIGURA 12: MODELO ENTIDADE-RELACIONAMENTO DO SUBSISTEMA OFERTA DE PRODUTOS E SERVIÇOS.	49
FIGURA 13: DIAGRAMA DE BANCO DE DADOS VERSÃO 20140407, DESENVOLVIDO COM A FERRAMENTA MYSQL WORKBENCH.	57
FIGURA 14: OCORRÊNCIA DE REDUNDÂNCIA DE DADOS.....	61
FIGURA 15: DIAGRAMA DE BANCO DE DADOS, GERADO PELA FERRAMENTA WEB PHPMYADMIN.	64

LISTA DE SIGLAS

DB	– <i>Database</i>
FTP	– Fundamentações Teóricas e Práticas
GUI	– <i>Graphics User Interface</i>
HTML	– <i>Hypertext Markup Language.</i>
ICH	– Interações Culturais e Humanísticas.
IDE	– <i>Integrated Development Environment.</i>
PA	– Projetos de Aprendizagem.
PHP	– <i>Personal Home Page Tools.</i>
SGBD	– Sistema de Gerenciamento de Banco de Dados.
SQL	– <i>Structured Query Language.</i>
UFPR	– Universidade Federal do Paraná.
UML	– <i>Unified Modeling Language.</i>
WWW	– <i>World Wide Web.</i>

SUMÁRIO

1 INTRODUÇÃO	14
2 JUSTIFICATIVA	16
2.1 MEMORIAL	16
2.2 UFPR LITORAL	18
2.3 PROBLEMATIZAÇÃO	19
3 OBJETIVOS	21
3.1 OBJETIVO GERAL	21
3.2 OBJETIVOS ESPECÍFICOS	21
4 FUNDAMENTOS TEÓRICOS	22
4.1 BANCO DE DADOS	25
4.2 FRAMEWORK	26
4.3 FERRAMENTAS UTILIZADAS	27
4.3.1 Sistema operacional Linux	27
4.3.2 Sistema Operacional Windows	28
4.3.3 MySQL Server/Client	30
4.3.3.1 A linguagem SQL	32
4.3.4 Microsoft Visio	34
4.3.5 Mozilla Firefox	35
4.3.6 PHPMyAdmin	37
4.3.7 MySQL Workbench	39
5 METODOLOGIA	41
6 DESENVOLVIMENTO DO SISTEMA	42
6.1 MODELAGEM DO SISTEMA	42
6.1.1 Plantas e Modelos Arquitetônicos	44
6.1.2 Fornecedores e Lojas de Materiais	45
6.1.3 Prestadores de Serviços, Profissionais Liberais e Autônomos	46
6.1.4 Mural de Serviços	48
6.2 VERSÕES DO BANCO DE DADOS E SUAS MODIFICAÇÕES	49
6.2.1 Versões	50
6.2.1.1 Versão 20131003	50
6.2.1.2 Versão 20140407	54

6.2.1.3 Versão 20140516	58
6.2.1.4 Versão 20140716	58
6.2.1.5 Versão 20140901	59
6.2.1.6 Versão 20140930	59
6.2.1.7 Versão 20141017	59
6.2.1.8 Versão 20141108	60
6.2.1.9 Versão 20141110	61
6.2.1.10 Versão 20150405	62
CONSIDERAÇÕES FINAIS.....	65
REFERÊNCIAS	67
APÊNDICES	69
APÊNDICE A: PORTFÓLIO: MODELO ENTIDADE-RELACIONAMENTO DO BANCO DE DADOS “ConstruCivil”	69
APÊNDICE B: CÓDIGO GERADO PELO PHPMYADMIN POR MEIO DO EXPORTADOR.....	70

1 INTRODUÇÃO

Neste trabalho, é descrito o processo de desenvolvimento de um sistema *WEB*, que é utilizado como ferramenta de cadastro e consulta por produtos e serviços voltados à construção civil no Litoral Paranaense.

Com os avanços na área de tecnologia da informação, ferramentas surgem para suprir a demanda por ferramentas de pesquisa. Porém, no litoral paranaense, não há nenhuma ferramenta capaz de suprir tal demanda, no que diz respeito à procura de profissionais e serviços na região. Acredita-se que a ferramenta proposta aqui pode contribuir para reduzir a peregrinação da população dos sete municípios da região, que são Antonina, Guaraqueçaba, Guaratuba, Matinhos, Morretes, Paranaguá e Pontal do Paraná.

Neste sentido, o sistema desenvolvido pode ser visto como um módulo de um projeto maior, que desenvolve uma página *Web*, e que utilize o sistema de banco de dados capaz de comportar essa ferramenta. Como proposta de solução, o projeto teve sua origem na vivência acadêmica, no convívio com os estudantes do curso de Informática e Cidadania, turma 2011, da UFPR Litoral e seus conhecimentos de mercado imobiliário na região.

Surgiu, assim, a ideia de construir uma ferramenta que sirva ao propósito de proporcionar uma melhor forma de buscar produtos e serviços no ramo de construção civil.

As ferramentas utilizadas neste projeto foram: o sistema gerenciador de banco de dados MySQL¹, a ferramenta *Web* de desenvolvimento de banco de dados

¹ MySQL: Sistema gerenciador de bancos de dados, responsável pelo armazenamento e processamento de informações. Disponível em: <<http://www.mysql.com>> acesso em: 31/05/2015.

PHPMyAdmin² e duas ferramentas de edição de diagramas UML³: Visio⁴ e LibreOffice Draw⁵.

Serão apresentados os diagramas de linguagem gráfica UML, criados com a ferramenta Visio e LibreOffice Draw, os atores e casos de uso básicos que geraram o início do desenvolvimento do sistema. O diagrama de classes, que deu origem aos *scripts*⁶ de MySQL que deram sentido para as primeiras utilidades ao funcionamento do sistema, realizado por meio da ferramenta *Web* PHPMyAdmin.

Por fim, verifica-se que os objetivos foram alcançados, tendo em vista que, conforme novas ideias, demandas e carência de meios de receber informações forem surgindo ao longo do tempo, novas necessidades surgem e geram novas versões para o mesmo sistema de base de dados.

² PHPMyAdmin: Ferramenta de desenvolvimento e gerenciamento de banco de dados em ambiente de página Web, cuja plataforma de programação é o MySQL. Disponível em: <http://www.phpmyadmin.net/home_page/index.php> Acesso em: 31/05/2015.

³ UML: Unified Modelling Language. Tradução nossa: linguagem de modelagem unificada.

⁴ Visio: Ferramenta de criação de diversos tipos de diagramas, desenvolvida pela Microsoft. Disponível em: <www.microsoftstore.com/Visio_2013> acesso em: 31/05/2015.

⁵ LibreOffice Draw: Software livre para criação de diagramas de diversos tipos. Disponível em: <<http://pt-br.libreoffice.org/>> acesso em 31/05/2015.

⁶ Script: programação executada no interior de outras linguagens de programação.

2 JUSTIFICATIVA

Tanto o sistema quanto este trabalho são tentativas de contribuir para eliminar a carência de informações sobre os prestadores de serviços e fornecedores de materiais, ambos voltados à construção civil na região, de forma prática e teórica, por meio da divulgação deste trabalho no meio acadêmico e seu uso em outras aplicações.

Este trabalho acadêmico descreve todas as dificuldades, modificações, acertos e resultados do banco de dados, por meio do processo de documentação de cada uma das versões.

2.1 MEMORIAL

Eu nasci em Paranaguá-PR em 1986. Meu primeiro contato com a informática foi em 1996, fazendo desenhos no aplicativo Paint, no Windows 95. Meu pai comprou um computador para uso pessoal em 1997. Mesmo meus pais separados, eu não deixei de visitá-lo. Meu pai não queria perder contato com os filhos. Eu queria ter contato mais frequente com o computador e interagir com ele o máximo possível.

Quando completei 18 anos, fiz um curso profissionalizante de informática, no qual aprendi a digitar e formatar textos, a trabalhar com planilhas eletrônicas, montar apresentações e falar em público, além de criar meu endereço de correio eletrônico. O curso durou um ano até eu receber o certificado.

Dois anos mais tarde, fiz outro curso relacionado à informática: manutenção de computadores. Pela primeira vez eu abri um gabinete e vi todo o *hardware* lá dentro. Aprendi que o CPU (*central processing unit*) não é o gabinete, é um microchip, que esquenta muito, tanto que precisa de um *cooler* para resfriá-lo. O curso durou quatro meses até receber o certificado.

Depois fui para Joinville-SC prestar vestibular para o curso de Sistemas de Informação. Mas, por falta de recursos financeiros, cursei apenas três anos, mesmo fazendo estágio remunerado como instrutor de informática para iniciantes em computadores fraquíssimos equipados com Linux. Em seguida, voltei para Paranaguá, na tentativa de continuar os estudos na área que eu mais gosto. Mas a mensalidade das universidades privadas só aumentava, e o poder aquisitivo da minha família era corroído pela inflação. Este fato, em conjunto com a completa ausência de postos de trabalho no litoral paranaense na área de informática, me fez decidir mudar de curso, cursei geografia durante alguns meses, outra área que eu sempre gostei, gosto de mapas e conhecer culturas de outros países. Eu continuaria lá, se não fosse pelo meu amigo Victor Hugo me avisar sobre o vestibular da UFPR, que tem um campus em Matinhos e que havia aberto um curso chamado Informática e Cidadania desde 2009. Fiz o vestibular, fui aprovado e iniciei meus estudos acadêmicos de informática.

Já cursando, conheci o Paiva, Gilson, Erick e Emerson, formando uma equipe. Conforme fomos estudando, tomamos conhecimento de um dos módulos ofertados pela universidade, os Projetos de Aprendizagem. Tínhamos que definir um tema de estudo, sendo aí que o Paiva pensou em algo que conciliaria informática e cidadania. Ele teve a ideia de montar uma página na Internet que reuniria informações relacionadas à construção civil, e assim minha contribuição seria importante, pois páginas de Internet necessitam de banco de dados, sendo uma área que penso ter conhecimento.

O Projeto de Aprendizagem foi dividido em duas equipes: uma para o desenvolvimento da página do Construtor Civil por Luciano e Erick e a outra equipe ficou responsável pela arquitetura do sistema, a cargo de Paiva, Gilson e eu.

Este trabalho acadêmico relata toda a minha pesquisa sobre o banco de dados que criei para o portal Construtor Civil.

2.2 UFPR LITORAL

A Universidade Federal do Paraná – Setor Litoral possui uma proposta pedagógica diferenciada, em relação às outras universidades públicas e privadas no Brasil. Tal proposta pedagógica baseia-se em projetos desenvolvidos junto às comunidades locais. A Universidade acredita que esta proposta pode contribuir e incentivar o desenvolvimento científico, econômico, cultural, ambiental e social no litoral paranaense.

A UFPR Litoral desenvolveu um projeto político-pedagógico (PPP), tendo como objetivo promover uma educação integrada conforme os anseios da comunidade litorânea, que busca, projetos que tenham como palavras-chave: sustentabilidade e social. Este PPP é composto por três eixos pedagógicos, que são: FTP⁷, ICH⁸ e PA⁹.

Segundo o PPP do Setor Litoral¹⁰, “o currículo fundamentado na educação por projetos estimula o estudante a construir os saberes, integrando diversas áreas do conhecimento”.

A UFPR Litoral foi inaugurada em 2004, pelo então governador Roberto Requião de Melo e Silva e pelo presidente Luís Inácio Lula da Silva. Quando entrou em funcionamento, foi firmado um pacto pelo desenvolvimento de um projeto de ensino que saísse da ortodoxia predominante no Brasil. Um exemplo disso é a organização dos assentos e mesas em sala de aula, normalmente voltados para a lousa, e a UFPR Litoral organizou os assentos virados de lado, formando um grande círculo. Acreditou-se que isto estimularia o convívio entre os outros acadêmicos e o diálogo. Segundo depoimentos de alunos já formados na UFPR Litoral, a

⁷ FTP: Fundamentações Teóricas e Práticas. Formam os módulos pedagógicos orientados ao como fazer, o que fazer e como atuar no cenário em questão, com teorias e práticas de atividades.

⁸ ICH: Interações Culturais e Humanísticas. “Através de encontros que ocorrem semanalmente, integrando estudantes dos diferentes cursos, o ICH constitui-se num espaço de aprendizagem interdisciplinar. Possibilita a articulação de diversos saberes (científicos, culturais, populares e pessoais) e busca um olhar mais amplo para a problemática cultural e humanística contemporânea”. UFPR Litoral.

⁹ PA: Projetos de Aprendizagem. Módulo pedagógico no qual o aluno desenvolve seus projetos como produtos, serviços ou conhecimento, de acordo com seus interesses, sob orientação de professores atuando como mediadores.

¹⁰ PPP: Projeto Político Pedagógico: Disponível em: <http://www.litoral.ufpr.br/portal/wp-content/uploads/2015/02/PPP-UFPR-LITORAL_Set-2008_Alteracao_Dez-2008.pdf> acesso em: 01/06/2015.

universidade não é só um caminho para a inserção no mercado de trabalho, mas também é um espaço onde é possível desenvolver uma trajetória de vida.

2.3 PROBLEMATIZAÇÃO

Os motivos que levaram a definir o tema são: carência de recursos na web, afinidade e prática com o tema no cenário Litoral do Paraná. Os saberes envolvidos serão basicamente os abordados durante o curso de informática e cidadania, sendo os principais: UML, Banco de Dados e *script* MySQL. Os espaços de aprendizagem, consistem basicamente em um escritório com servidor de rede e um computador conectado à Internet, mesmo que com acesso remoto.

Houve a necessidade de averiguar onde estão e quantos são os prestadores de serviços, como pedreiro e servente, por exemplo, em uma cidade da região, como Matinhos ou Paranaguá, e qual a forma de contatá-los. A primeira questão tinha seu domínio na consulta por localidade. O que tem de produtos para construção civil em Guaraqueçaba? Quais são os serviços disponíveis nessa cidade? Um pedreiro de Paranaguá pode querer procurar um trabalho em Guaraqueçaba. Mas tem serviço para ele lá? A segunda questão é consultar imóveis por padrão de construção. Qual é o imóvel disponível em Guaratuba, cujo valor é menor que R\$ 200.000,00? Existe algum imóvel ecológico sustentável em Morretes ou em Antonina? A terceira questão é cadastrar prestadores de serviços e classificá-los. Quantos arquitetos têm aqui? Como saber quantos são os pedreiros em Paranaguá? Como entrar em contato com um deles? Tem algum jardineiro em Guaraqueçaba? Qual o endereço dele? Por fim, a quarta questão é procurar e cadastrar os fornecedores de materiais. Onde é mais barato comprar um pacote de cimento em Pontal do Paraná? Qual produto está em oferta em uma loja de materiais de construção em Paranaguá?

Antes do início do desenvolvimento do banco de dados, foi pensado em uma possibilidade de fazer uma versão não-relacional em paralelo com o tradicional, banco de dados relacional. Está em ascensão no mercado, a tecnologia que trabalha com bancos de dados não-relacionais, e já é utilizado por buscadores como a

Google¹¹, para tratar de grandes quantidades de informações. Este tipo de banco de dados, não-relacional, é conhecido como NoSQL¹². De acordo com Sadalage e Fowler (2013, p. 14), o modelo de banco de dados não relacional, foi criado a partir da “necessidade de manipulação de volumes maiores de dados, o que impôs uma mudança fundamental no sentido de desenvolvimento de plataformas grandes de *hardware* por meio de *clusters* de servidores comerciais”.

Por hora, a prioridade foi o modelo relacional, em função do conhecimento adquirido acerca do NoSQL não ser suficiente para aplicar sobre o sistema de banco de dados para o portal Construtor Civil.

¹¹ Google: Portal de pesquisas da Internet, empresa multinacional de serviços de Internet, sediada na cidade de Mountain View, Califórnia. Disponível em: <<http://www.google.com>> acesso em: 01/06/2015.

¹² NoSQL: Termo utilizado para definir bancos de dados não-relacionais. Também é interpretado como Not Only SQL. Disponível em: <<http://pt.slideshare.net/RodrigoDeSouzaValerio/bancos-de-dados-nosql-not-only-sql>> Acesso em: 07/06/2015.

3 OBJETIVOS

3.1 OBJETIVO GERAL

O objetivo deste trabalho é descrever o desenvolvimento de um sistema de base de dados, cuja finalidade é reunir informações de usuários do portal eletrônico Construtor Civil¹³. Com isso, pretende-se integrar o portal eletrônico ao banco de dados, onde serão cadastrados os prestadores de mão-de-obra em todas as categorias, além de fornecedores de materiais.

Para isto, é necessário abordar as tecnologias utilizadas para o desenvolvimento da solução.

3.2 OBJETIVOS ESPECÍFICOS

Como objetivos específicos do sistema, pretende-se:

- Levantar os requisitos de um sistema de banco de dados
- Criar um mapa conceitual acerca do sistema em questão
- Elaborar os Casos de Uso do sistema
- Desenvolver o banco de dados ConstruCivil
- Testar e avaliar o banco de dados ConstruCivil

¹³ Construtor Civil. Portal eletrônico dedicado a serviços de busca por produtos e serviços na área da construção civil no Litoral Paranaense. Disponível em: <<http://www.construtorcivil.com.br>> Acesso em: 31/05/2015.

4 FUNDAMENTOS TEÓRICOS

O primordial do desenvolvimento do banco de dados é evitar que informações previamente cadastradas sejam digitadas diversas vezes e de variadas formas (mal escritas ou abreviadas), pelo usuário final, em tabelas diferentes que nada têm a ver com o contexto requerido. Isso pode causar redundâncias, imprecisões e falta de coesão das informações.

Redundância, segundo Heuser (2008, p. 20), “ocorre quando uma determinada informação está representada no sistema várias vezes (...), informações referentes a um produto que aparecem nos arquivos de produtos de cada um dos sistemas”. O autor sugere que pode haver vários sistemas, ou tabelas, e uma informação estar representada várias vezes em diversas tabelas. Logo, a importância de se condicionar os dados, isto é, cadastrar determinadas informações, de forma que o usuário final não precise digitar outras vezes a mesma informação e que essa informação esteja na tabela correta do banco de dados.

Envolve também, cadastrar informações corretamente, sem abreviações nem erros de ortografia, em tabelas-chave, fazer com que o usuário escolha uma informação em uma lista, que receberá as informações dessa tabela-chave. Por exemplo, existe uma tabela no banco chamada “Clientes”. Um usuário X digitaria a palavra “Paranaguá” no campo “Cidade”. Outro usuário chamado Y digitaria “Paranagua” (sem acento) no mesmo campo. Mas um outro usuário chamado Z pode digitar nesse mesmo campo a informação “Pgua” ou “Pguá”, formas abreviadas de se referir à cidade de Paranaguá. Quando se fazer necessário procurar clientes cuja cidade é Paranaguá, não serão listados todos os clientes, pois cada cliente digitou o nome da cidade de forma diferente, o que compromete a busca pelos clientes por cidade e, em consequência, a credibilidade e a consistência do banco de dados. Então, quando se fala em condicionar os dados, fala-se em padronizar as informações e separá-las por tabela, de acordo com o interesse. No caso deste exemplo, foi decidido criar uma tabela chamada “Cidades”, onde estão cadastrados o código de identificação, o nome da cidade e a unidade federativa (UF) a qual está situada. Essa prática se chama “Normalização de dados”. Battisti (2015,

pg. 1) explica que a normalização, baseada em formas normais “é uma regra de deve ser aplicada na construção das tabelas do banco de dados para que estas fiquem bem projetadas”. Esta prática serve para que um registro seja feito de um jeito só e caberá ao usuário escolher a cidade já cadastrada na tabela “Cidades”.

A necessidade de se realizar tal prática vem da ideia de simplificar o processo de distribuição das informações e da alimentação do banco de dados. Machado (2008, p. 34) complementa que o objetivo de se normalizar, ou padronizar, as informações é “evitar os problemas que podem provocar falhas no projeto de banco de dados, bem como **eliminar a mistura de assuntos** e as correspondentes redundâncias desnecessárias de dados”. Padlipskas (2015, pg. 1) complementa:

Muitas das empresas que não possuem esse tipo de metodologia implementada, constantemente buscam novas práticas de gerenciamento para diminuir seus custos diretos de desenvolvimento e manutenção de software.

Padlipskas (2015) demonstra que é necessário dar uma nova versão ao banco de dados a cada modificação, que é realizada na estrutura de dados e no relacionamento entre as tabelas. A seguir, para fins de conveniência, será substituída a expressão “banco de dados” pela abreviatura em inglês DB (*database*).

As consequências de não organizar a alimentação e a distribuição dos dados inseridos no DB são apontados por Milani (2006, p. 328):

A má estruturação de um banco de dados pode gerar diversos problemas para sua aplicação e lógica de negócio, podendo tornar o sistema para o qual o banco de dados trabalha menos preciso, lento, incoerente, redundante ou, até mesmo, não confiável.

No primeiro momento de preparar a primeira versão do DB “ConstruCivil”, não havia sido pensado em uma forma de normalizar os dados, apenas em separá-los por área de interesse (cidade para a tabela Cidades e serviço específico para a tabela Tipos de Serviços, etc.). Mas era de conhecimento que a necessidade de fazê-lo aumentaria na medida em que a dinâmica crescesse. Com isso, chegaria o momento de descrever as maneiras de como a dinâmica fez o DB crescer até atingir certo ponto e as maneiras de padronizar a alimentação e a distribuição de

informações nas tabelas. Então, surgiu a ideia de se documentar cada versão do DB “ConstruCivil”.

Heuser (2008, p. 20) ajuda a definir a forma como o banco de dados “ConstruCivil” foi formado, conforme cada versão desenvolvida até chegar ao estado atual:

Muitas vezes, a implantação da informática em organizações ocorre de forma evolutiva e gradual. Inicialmente, apenas determinadas funções são automatizadas. Mais tarde, à medida que o uso da informática vai se estabelecendo, novas funções vão sendo informatizadas.

“Esse conteúdo tende apenas a crescer. Um gestor de dados deve entender sobre o armazenamento de informações virtuais e saber fazer uma análise correta sobre este conteúdo”, é o que diz um artigo do portal Globo Educação¹⁴ (2015), citação esta que ajuda a entender como é a dinâmica de uma montagem de um banco de dados. A tarefa de um gestor de banco de dados é entender a dinâmica de uma empresa, quais são as informações que são trabalhadas diariamente e montar uma ferramenta capaz de colher resultados relevantes ao negócio.

Ainda sobre documentação, Guimarães (2013, pg 1) fala sobre a importância da documentação de um projeto, no caso deste trabalho acadêmico, é um projeto de banco de dados para um portal na Internet orientado ao auxílio de procura por serviços no ramo da construção civil:

Os projetos são, por sua natureza, realizados de forma única para criar produtos, serviços e outros resultados que, por sua vez, também são únicos. Em função disso, muitos gestores acreditam que as lições aprendidas no passado não podem ser aplicadas no futuro. Apesar dessa constatação, é preciso lembrar que todos os projetos possuem elementos em comum que devem ser considerados na fase de documentação.

O DB “ConstruCivil” deve ser capaz de comportar adequadamente todas as informações pertinentes à realidade do mercado de construção civil voltado para o público do Litoral Paranaense, de forma que os usuários possam cadastrar ou apontar eventuais erros quanto à realidade local, de forma colaborativa. Prestadores de serviços, profissionais liberais e autônomos, lojas de materiais de construção, enfim, todos os envolvidos na área de prestação de serviços e de arquitetura e

¹⁴ Globo Educação. Disponível em: <<http://educacao.globo.com/artigo/profissoes-do-futuro-tecnologia-movimenta-mercado-de-trabalho.html>> acesso em 03/04/2015.

construção, que são o público-alvo que o sistema proposto pretende atingir. Fontenelle (2015, pg. 1) explica que:

Todo ambiente pode transmitir uma mensagem e se posicionar. No mercado atual, mais do que simplesmente produtos e serviços, o público deseja três coisas: uma identidade clara que expresse a alma da empresa e de seus produtos/serviços, valores com os quais ele se identifique e coerência na concretização desse discurso.

4.1 BANCO DE DADOS

Entende-se por banco de dados, como um modelo estruturado de processamento de informações, de forma a organizar e condicionar os dados segundo critérios apurados no modelo de negócio, o qual se propõe um projeto. De acordo com Machado (2008, p.20), banco de dados é “um conjunto de dados devidamente relacionados (...), objetos conhecidos que podem ser armazenados e que possuam um significado implícito”. O mesmo autor (2008, p.4) define bancos de dados relacionais como “conjuntos de dados vistos segundo um conjunto de tabelas, e as operações que as utilizam são realizadas por linguagens que o manipulam, não sendo procedurais, ou seja, manipulados de uma só vez”. A linguagem, a qual o autor se refere, atende pela sigla em inglês SQL.

O banco de dados utilizado neste projeto foi denominado como “ConstruCivil”. Será a estrutura de dados montada para receber as informações do usuário final, que alimenta banco por meio de formulários de Internet, a serem implementados no futuro portal eletrônico Construtor Civil. Silva (2011, p. 196) explica que formulário é “a funcionalidade de integração com o usuário mais usada na *web* e também uma das mais antigas. Sua função é recolher dados entrados pelo usuário e enviá-los ao servidor para processamento”. O mesmo autor complementa que o usuário preenche um formulário e envia todas as informações digitadas nos campos da página *web* para o processamento por um agente que, neste caso, é o banco de dados que será transferido para um servidor de hospedagem. As empresas mais conhecidas no Brasil que trabalham com hospedagem de dados são:

Hostnet¹⁵ e Locaweb¹⁶.

De acordo com Heuser (2008, p. 184), “raramente, os arquivos destes sistemas estão documentados por meio de modelos conceituais”. O mesmo autor ainda completa: “Adicionalmente, há bancos de dados conceituais que não possuem documentação na forma de um modelo conceitual”. Porém, no caso do banco de dados “ConstruCivil”, o mesmo autor faz uma ressalva: “No entanto, há situações no ciclo da vida de um sistema nas quais um modelo conceitual é de grande valia”.

4.2 FRAMEWORK

Segundo Fayad e Schmidt (1997, pg. 1), *frameworks* são aplicações orientadas a objetos e são “uma tecnologia promissora para usar projetos e execuções de softwares testados a fim de reduzir o custo e melhorar a qualidade do *software*”. E complementam os autores, “um conjunto de classes que colaboram para realizar uma responsabilidade para um domínio de um subsistema da aplicação”. *Framework* pode ser considerado um ambiente visual de um sistema operacional. O sistema operacional *Windows*¹⁷ utiliza seu próprio *framework*, o “Microsoft .NET Framework”, que é de uso exclusivo em sistemas operacionais desenvolvidos pela Microsoft, para desenvolvimento de sistemas e aplicações. Este *framework* foi requisitado para a instalação da ferramenta “MySQL Workbench”¹⁸, no sistema operacional “Windows”.

Sistemas operacionais Linux¹⁹ são de uso livre e gratuito, diferentemente do Windows, que é pago. Os sistemas operacionais Linux possuem diversos *frameworks*, que variam de acordo com a distribuição (ou *distro*, como costumeiramente utilizado pelos desenvolvedores). São exemplos de *frameworks*:

¹⁵ Hostnet: Servidor de hospedagem de sites, banco de dados e arquivos em geral. Disponível em: <<http://www.hostnet.com.br>>, acesso em 31/05/2015.

¹⁶ Locaweb: Servidor de hospedagem de sites, banco de dados e arquivos em geral. Disponível em: <<http://www.locaweb.com>>, acesso em 31/05/2015.

¹⁷ Windows: sistema operacional desenvolvido pela empresa norte-americana Microsoft. Disponível em: <<http://www.microsoft.com/pt-br/default.aspx>> acesso em: 31/05/2015.

¹⁸ MySQL Workbench: ferramenta GUI para criação de banco de dados do SGDB MySQL. Disponível em: <<https://www.mysql.com/products/workbench/>> acesso em: 01/06/2015.

¹⁹ Linux: sistema operacional de código-fonte aberto à comunidade de desenvolvedores. Disponível em: <<http://www.canonical.com/>> acesso em 31/05/2015.

KDE, Gnome, Unity, Cinnamon e XFCE. A distro utilizada no desenvolvimento do banco de dados “ConstruCivil” chama-se Ubuntu, que utiliza a *framework* Unity.

4.3 FERRAMENTAS UTILIZADAS

4.3.1 Sistema operacional Linux

Linux é um sistema operacional como o Windows, que é o mais utilizado e mais conhecido. Linux possui várias *distros*²⁰ de acordo com o desejo de seus desenvolvedores. Há *distros* mais simples, do mesmo modo que há *distros* mais sofisticados.

O desenvolvimento do banco de dados “ConstruCivil” foi desenvolvido tanto em Linux como no *Windows*. O Ubuntu²¹ foi a *distro* mais frequentemente utilizada dentre as *distros* Linux para desenvolvimento de aplicativos e sistemas, juntamente com a Linux Mint²², que é mais voltada para uso pessoal para fins de entretenimento e multimídia.

Foi optado utilizar a *distro* Ubuntu (FIGURA 1), pelo elevado grau de maturidade entre as várias *distros* Linux existentes no mercado. Fora o fato de que o uso do Linux é livre e gratuito, graças à licença GPL (*General Public License*)²³, licença para *software* livre, a mais utilizada para permitir que qualquer usuário pode utilizar. Inclusive, é possível iniciar o funcionamento do computador por meio do processo chamado *dual-boot*, onde o usuário pode escolher entre utilizar o Linux ou o *Windows*, assim que liga o computador, fato que ganhou simpatia entre os usuários que adotam o Linux, uma vez percebida que não são todos os aplicativos *Windows* que rodam em Linux.

²⁰ *Distro*: termo utilizado pelos desenvolvedores de software livre para designar uma distribuição Linux, com propósitos específicos e voltado para um grupo de usuário. Disponível em: <www.inf.ufsc.br/~bosco/extensao/sl-conceitos_basicos_de_linux.pdf> acesso em: 31/05/2015.

²¹ Ubuntu: *distro* desenvolvida pela empresa Canonical. Disponível em: <<http://www.ubuntu.com/>> acesso em: 31/05/2015.

²² Linux Mint: *distro* desenvolvida por diversos desenvolvedores Linux, cujo código-fonte é proveniente do Ubuntu. Disponível em: <<http://www.linuxmint.com.br/>> acesso em: 31/05/2015.

²³ GNU GPL: Licença a qual permite que qualquer pessoa “copie e distribua cópias, sem alterações no documento de licença, sendo vedada qualquer modificação”. Disponível em: <<http://creativecommons.org/licenses/GPL/2.0/legalcode.pt>> acesso em: 01/07/2015.

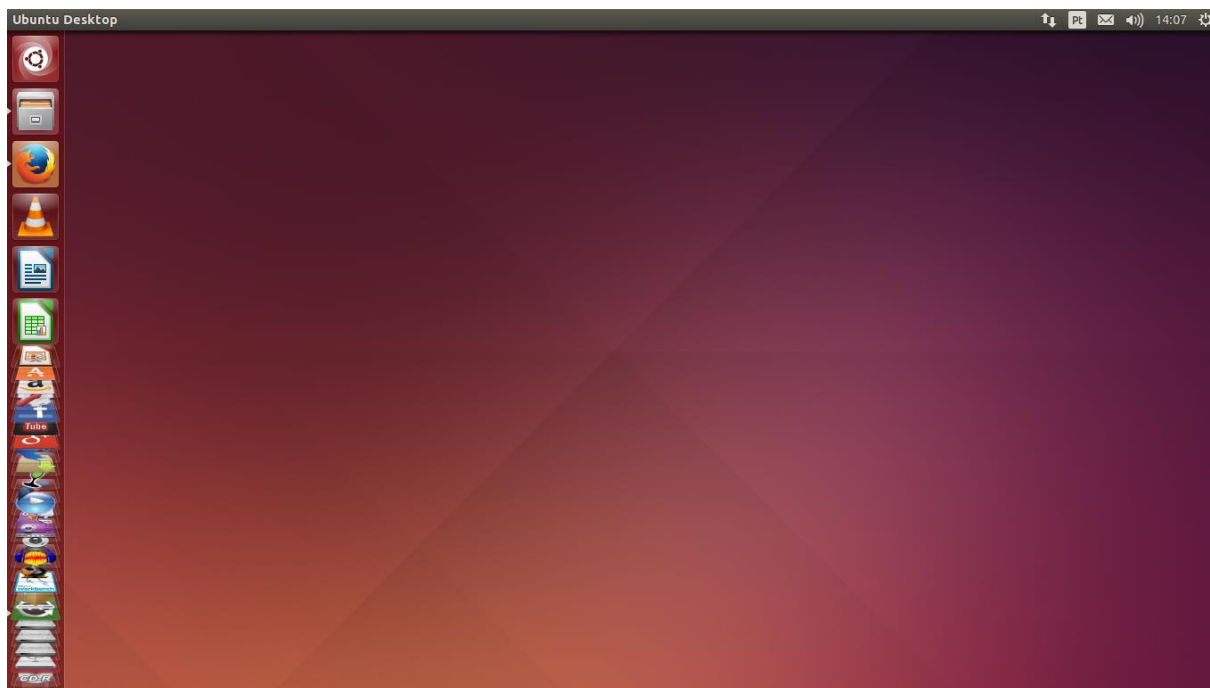


FIGURA 1: TELA INICIAL DO LINUX – DISTRO UBUNTU VERSÃO 14.04 LTS 64 BITS.
FONTE: O Autor (2015).

4.3.2 Sistema Operacional *Windows*

Windows é o sistema operacional mais conhecido e mais utilizado no mercado, tanto entre usuários domésticos quanto entre os desenvolvedores. É um produto comercial, cujos preços variam de acordo com a configuração do sistema. O mais simples é a configuração *Home*²⁴, cujo preço atualmente gira em torno de R\$ 250,00 e possui configurações dedicadas a usuários domésticos. Possui a versão *Pro*²⁵, cujo preço gira em torno de R\$600,00 e é apropriado para desenvolvedores de sistemas.

²⁴ *Windows 8 Home*: versão com configurações básicas. Disponível em: http://www.microsoftstore.com/store/msbr/pt_BR/pdp/productID.288436900> Acesso em: 06/06/2015.

²⁵ *Windows 8 Pro*: versão voltada para desenvolvedores e usuários de negócios, com suporte à computação remota. Disponível em: http://www.microsoftstore.com/store/msbr/pt_BR/pdp/productID.288437100> acesso em: 06/06/2015.

Tanto é que a noção de seu funcionamento, contexto, prática, experimento e grande número de aplicativos compatíveis faz com que este sistema operacional se torne indispensável, principalmente entre os leigos em informática.

A versão mais recente do *Windows* é a 8.1 (FIGURA 2), que reúne o útil ao agradável, com imagens limpas, várias opções de personalização, aliados à confiabilidade quanto a quesitos como desempenho, estabilidade, consumo de memória, conectividade e portabilidade ao rodar aplicativos, como os desenvolvidos em 32 bits que rodam em versão do sistema operacional de 64 bits, que permite utilização de toda a capacidade de memória processada.



FIGURA 2: TELA INICIAL DO WINDOWS 8.
FONTE: O Autor (2015).

4.3.3 MySQL Server/Client

MySQL é um sistema gerenciador de banco de dados (SGBD) que utiliza a linguagem SQL como linguagem de consulta. Por ser uma ferramenta *open-source*²⁶, é um dos SGBDs mais famosos do mundo. Segundo informações do portal DB-Engines, há mais de 10 milhões de usuários no mundo todo. Grandes empresas como Google, Cisco²⁷, HP²⁸ e NASA²⁹ são os usuários mais conhecidos deste SGBD. O banco Bradesco é o mais famoso entre as empresas brasileiras a adotar o

²⁶ Open-source: do inglês: código aberto. Refere-se a software livre. Disponível em: <<http://opensource.org/>> Acesso em: 06/06/2015.

²⁷ Cisco: Empresa multinacional de soluções para redes e comunicações, fabricação e venda de ferramentas de conexão de Internet, sediada na cidade de San José, Califórnia. Disponível em: <<http://www.cisco.com/web/BR/index.html>> acesso em: 01/06/2015.

²⁸ HP: Hewlett Packard, empresa norte-americana especializada em soluções de impressão, tratamento de imagens, software e serviços relacionados a fotografia, sediada na cidade de Palo Alto, Califórnia. Disponível em: <<http://www8.hp.com/br/pt/home.html>> acesso em: 01/06/2015.

²⁹ NASA: National Aeronautics and Space Administration, agência estatal norte-americana responsável pela pesquisa e desenvolvimento de tecnologias e programas de exploração espacial. Disponível em: <<http://www.nasa.gov>> acesso em: 01/06/2015.

MySQL. A Dataprev³⁰ é uma empresa estatal federal brasileira, vinculada ao Ministério da Previdência Social, e tem como SGBD o MySQL. São exemplos de sistemas criados pela estatal, utilizando o MySQL: SGA³¹, que funciona, de acordo com informações disponibilizadas no site da empresa estatal, “funciona exclusivamente em interface web. Construído em linguagem de programação PHP, banco de dados MySQL e servidor Web Apache³², o SGA utiliza também servidores com sistema operacional GNU/Linux³³ Debian³⁴”; e SIPREV³⁵, apresentado pela primeira vez no primeiro dia do I Encontro Nacional do *Software* Público, em Brasília. Esse sistema foi desenvolvido pela Unidade de Desenvolvimento do Ceará, uma das cinco unidades em território nacional. A base de dados é realizada com o SGBD MySQL, em conjunto com o ambiente de máquina virtual e linguagem de programação Java³⁶.

O funcionamento do MySQL é totalmente procedural, ou seja, não dispõe de GUI para realizar qualquer operação.

³⁰ Dataprev: Empresa de Tecnologia e Informações da Previdência Social. Disponível em: <<http://portal.dataprev.gov.br/>> acesso em: 01/06/2015.

³¹ SGA: Sistema de Gerenciamento de Atendimento, criado pela Dataprev.

³² Apache: Servidor de rede compatível com protocolo HTTP para transmissão via Web, processamento de dados e execução de aplicativos distribuídos. Disponível em: <<http://www.apache.org/>> acesso em: 01/06/2015.

³³ GNU/Linux: denominação de um sistema operacional que tem como código-fonte a linguagem UNIX.

³⁴ Debian: Distribuição Linux. Disponível em: <<http://www.debian.org/index.pt.html>> acesso em: 01/06/2015.

³⁵ SIPREV: Sistema Previdenciário de Gestão de Regimes Públicos de Previdência Social. Disponível em: <<http://www.previdencia.gov.br/siprevgestao-rpps-sistema-previdenciario-de-gestao-de-rpps/>> acesso em: 01/06/2015.

³⁶ Java: Linguagem de programação orientada a objetos compilada em máquina virtual, desenvolvida pela empresa Sun Microsystems, na década de 1990. Disponível em: <http://java.com/pt_BR/> acesso em: 01/06/2015.

```

thiago@thiago-GA-B75M-D3H: ~
Your MySQL connection id is 96
Server version: 5.5.41-0ubuntu0.14.04.1 (Ubuntu)

Copyright (c) 2000, 2014, Oracle and/or its affiliates. All rights reserved.

Oracle is a registered trademark of Oracle Corporation and/or its
affiliates. Other names may be trademarks of their respective
owners.

Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

mysql> use constructivil
Reading table information for completion of table and column names
You can turn off this feature to get a quicker startup with -A

Database changed
mysql> show tables
+-----+
| Tables_in_constructivil |
+-----+
| cidades                 |
| clientes                 |
| estados                 |
| fale-conosco            |
| fornecedores-de-materiais |
| mural-de-servicos       |
| ofertas-do-dia          |
| padroes-de-construcao   |
| plantas-e-modelos-arquitetonicos |
| prestadores-de-servicos |
| profissionais           |
| ranking-das-lojas       |
| reputacao               |
| tipos-de-avaliacao      |
| tipos-de-materiais      |
| tipos-de-servicos       |
+-----+
16 rows in set (0.00 sec)

mysql>

```

FIGURA 3: DEMONSTRAÇÃO DE FUNCIONAMENTO DO MYSQL CLIENT.

FONTE: O Autor (2015).

4.3.3.1 A linguagem SQL

SQL é uma linguagem de consulta de informações, assim como diz o nome, de origem inglesa, que significa Linguagem de Consulta Estruturada. Foi inicialmente desenvolvida pela empresa IBM³⁷ e tornou-se uma linguagem comercial para consultas em bancos de dados, com a popularidade que teve a empresa naquela época. Daí, foi necessária uma padronização da linguagem e foi, ela, organizada pelas associações internacionais ANSI³⁸ e ISO³⁹, equivalentes a órgãos brasileiros, como INMETRO⁴⁰ e ABNT⁴¹, citados aqui apenas para fins de comparação.

³⁷ IBM: Empresa norte-americana de produtos e soluções em informática, como hardware, software, serviços de infra-estrutura e de consultoria, hospedagem de páginas Web e pesquisas sobre a Nanotecnologia. Disponível em: <<http://www.ibm.com/>> acesso em: 01/06/2015.

³⁸ ANSI: American National Standards Institute. Disponível em: <<http://www.ansi.org/>> acesso em: 01/06/2015.

³⁹ ISO: International Organization for Standardization. Disponível em: <<http://www.iso.org/iso/home.html>> acesso em: 01/06/2015.

⁴⁰ INMETRO: Instituto Nacional de Metrologia, Qualidade e Tecnologia, autarquia estatal federal brasileira. Disponível em: <<http://www.inmetro.gov.br/>> acesso em: 01/06/2015.

Apesar da padronização ter sido adotada para a universalização da linguagem, Lima (2004, pg 1), afirma:

“Existem variações da linguagem (chamadas de extensões) que foram incorporadas para complementar as capacidades da linguagem. Tais variações podem comprometer a portabilidade, pois para obtê-la, o padrão deve ser rigorosamente seguido, não utilizando algumas facilidades incorporadas nas extensões”.

E isso explica como existem vários padrões adotados por fabricantes de *software* diferentes. Por exemplo, a sintaxe adotada pela *Microsoft*, com o seu produto *SQL Server* foi definida da seguinte maneira, para criar uma tabela de banco de dados chamada “Cidades”:

```
create table Cidades (
  "ID" int (10) not null primary key,
  "Nome_Cidade" char (50) not null,
  "Estado" char (2) not null
)
```

E a sintaxe do MySQL é diferente para se fazer a mesma coisa:

```
CREATE TABLE IF NOT EXISTS `Cidades` (
  `ID` int(10) NOT NULL,
  `Nome_Cidade` varchar(200) NOT NULL,
  `Estado` varchar(2) NOT NULL,
  PRIMARY KEY (`Código da Cidade`),
  KEY `Nome da Cidade` (`Cidade-UF`)
) ENGINE=InnoDB DEFAULT CHARSET=latin1;
```

Portabilidade, como disse Lima, é, realmente, um problema quando se deseja migrar um banco de dados de uma suíte para outra, bem como do *SQL Server* para o *MySQL* ou vice-versa. Então, a comunidade de desenvolvedores do *MySQL* desenvolveu uma ferramenta de migração para aqueles que criaram bancos

⁴¹ ABNT: Associação Brasileira de Normas Técnicas. Disponível em: <<http://www.abnt.org.br/>> acesso em: 01/06/2015.

de dados no ambiente *SQL Server* para o *MySQL*. O objetivo da empreitada foi trazer mais DBAs para a ferramenta *open-source*. E isso contribuiu para o crescimento e a fama deste SGBD.

6.3.4 Microsoft Visio

Microsoft Visio (FIGURA 4) é uma ferramenta para criação de diagramas, fluxogramas, modelagem de dados e esquemas de funcionamento e de organização social, como hierarquias, organogramas, cronogramas, que opera somente no sistema operacional *Windows*. Foi desenvolvido, inicialmente, pela finada empresa Visio Corporation⁴², uma empresa independente que foi adquirida em 2000 pela Microsoft. Desde então, o Visio foi incorporado ao pacote de ferramentas de produtividade. É concorrente direta de ferramentas idênticas como a “AutoCAD”⁴³.

⁴² Visio Corporation: foi uma empresa de software, cujo produto principal era um software de aplicativo de diagramação chamado Visio. Disponível em:

<<http://www.crunchbase.com/organization/visio-corporation>> acesso em: 06/06/2015.

⁴³ AutoCAD: Aplicativo de desenho tridimensional auxiliado por computador, criado pela empresa Autodesk em 1982. Disponível em: <<http://www.autodesk.com.br/products/autocad/overview>> acesso em: 01/06/2015.

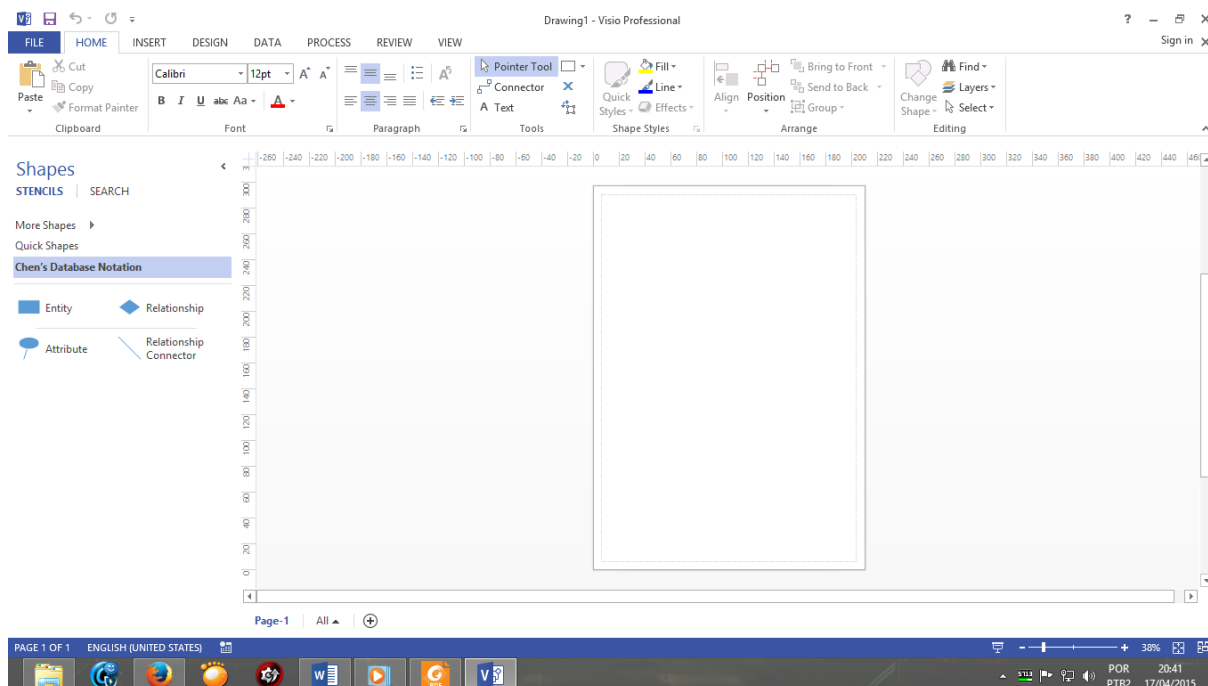


FIGURA 4: MICROSOFT VISIO RODANDO NO SISTEMA OPERACIONAL WINDOWS 8.
FONTE: O Autor (2015).

4.3.5 Mozilla Firefox

O Firefox (FIGURA 5) é um navegador de Internet e Intranet, desenvolvido pela *Mozilla Foundation*⁴⁴, e pode ser instalado em sistemas operacionais Linux, Windows e MacOS⁴⁵, como também em dispositivos portáteis como *smartphones*⁴⁶ e *tablets*⁴⁷, e em todos os sistemas operacionais a eles dedicados, como *Android*⁴⁸, *IOS*⁴⁹ e *Windows Phone*⁵⁰, sendo, portanto, um navegador multi-plataforma. É

⁴⁴ Mozilla Foundation: Organização sem fins lucrativos que mantém ferramentas open-source, como o Firefox e o Thunderbird, além de complementos para os mesmos. Disponível em: <<http://www.mozilla.org/pt-BR/about/>> acesso em: 01/06/2015.

⁴⁵ MacOS: Sistema Operacional desenvolvido pela empresa norte-americana Apple Inc. Funciona somente em computadores desenvolvidos pela mesma, chamados Macintosh. Disponível em: <<http://www.apple.com/br/osx/>> acesso em: 01/06/2015.

⁴⁶ Smartphone: Telefone celular com funcionalidades extensíveis via programas executados pelo sistema operacional instalado no mesmo, como se fosse um computador.

⁴⁷ Tablet: Dispositivo microeletrônico em formato de prancheta, que pode ser usado, como se fosse um microcomputador, para acesso à Internet, visualização de fotos, vídeos, leitura de livros, jornais e revistas e para entretenimento em geral.

⁴⁸ Android: Sistema operacional open-source para smartphones e tablets, baseado no núcleo Linux, desenvolvido pela empresa Google. Disponível em: <<http://www.android.com/>> acesso em: 01/06/2015.

⁴⁹ IOS: Também conhecido como iPhone OS, é um sistema operacional para smartphones iPhone desenvolvido pela Apple Inc. Disponível em: <<http://www.apple.com/br/ios/>> acesso em: 01/06/2015.

herdeiro da extinta Netscape⁵¹. Com o fim desta, foi criada a fundação *Mozilla* (*Mozilla Corporation*), que é formada por colaboradores nos cinco continentes, dedicados ao desenvolvimento de um navegador leve, seguro, intuitivo e extensível, requisitos estes muito visados entre os consumidores. Estima-se que 40% do código-fonte foi escrito por voluntários, fato este que contribuiu na popularização do aplicativo. É concorrente direto do navegador *Chrome*⁵², desenvolvido pela norte-americana *Google*. Juntos, esses dois navegadores diminuíram quase que totalmente a participação do navegador *Internet Explorer*⁵³, que vem instalado no sistema operacional *Windows*, pela incapacidade de lidar com complementos e extensões, coisas estas que o *Firefox* e o *Chrome* já dominam desde o lançamento de suas primeiras versões.

⁵⁰ Windows Phone: Sistema operacional para smartphones e tablets desenvolvido pela empresa Microsoft. Disponível em: <<http://www.windowsphone.com/pt-br>> acesso em: 01/06/2015.

⁵¹ Netscape: Empresa de serviços de Internet, software e telecomunicações, conhecida pelo navegador de Internet do mesmo nome. Era subsidiária da extinta America On-Line (AOL). Disponível em: <<http://isp.netscape.com/>> acesso em: 01/06/2015.

⁵² Chrome: Navegador de Internet desenvolvido pela Google. Disponível em: <<http://www.google.com/chrome/>> acesso em: 01/06/2015.

⁵³ Internet Explorer: Navegador de Internet desenvolvido pela Microsoft. Disponível em: <<http://windows.microsoft.com/pt-br/internet-explorer/download-ie>> acesso em: 01/06/2015.

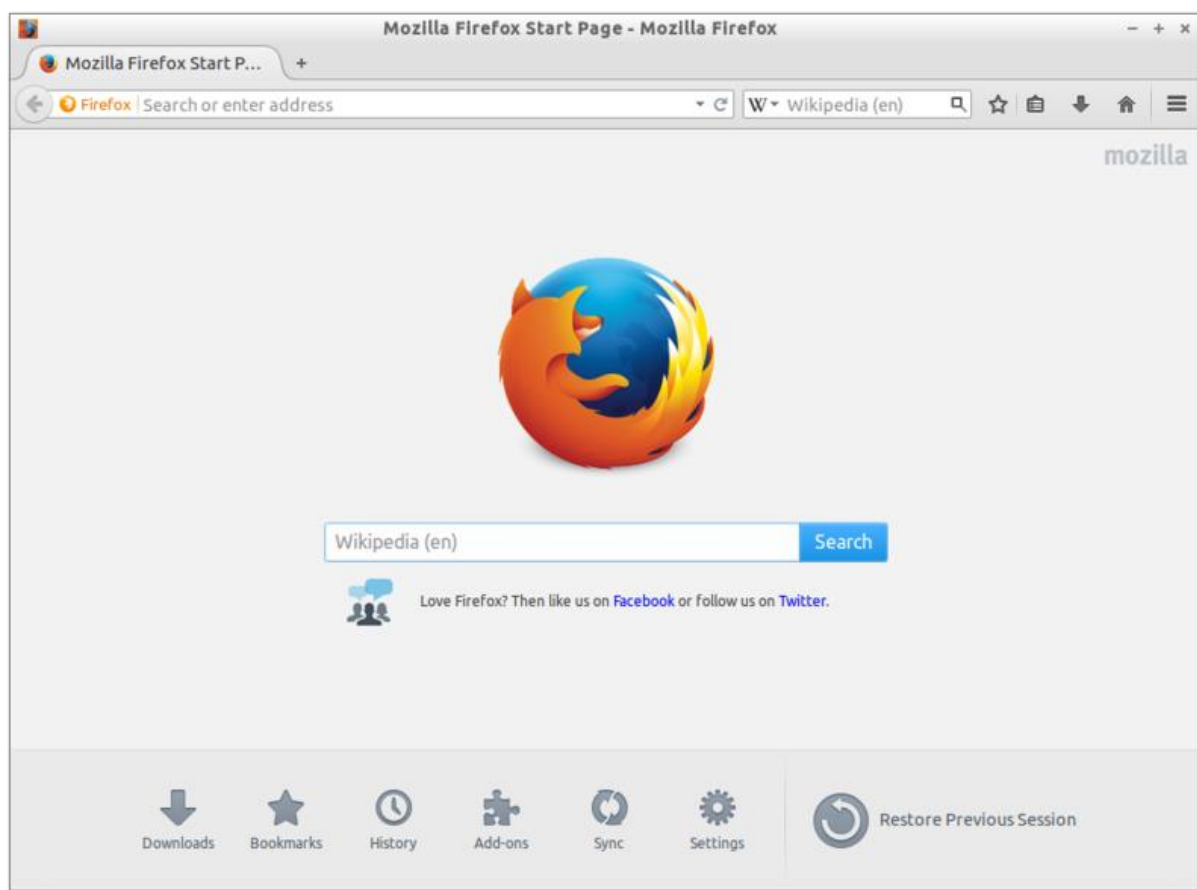


FIGURA 5: MOZILLA FIREFOX.
FONTE: O Autor (2015).

4.3.6 PHPMyAdmin

PHPMyAdmin é um aplicativo Web desenvolvido na linguagem de programação PHP, para administrar banco de dados e tem como matriz o SGBD MySQL. Foi criado pelo desenvolvedor e consultor de TI alemão Tobias Ratschiller. Ele criou ferramentas similares com outros SGBDs como matriz de programação, como o PHPMSAdmin⁵⁴, ferramenta que utiliza como matriz de programação o

⁵⁴ PHPMSAdmin: Administrador de banco de dados via página Web com SQL Server, SGBD da Microsoft. Disponível em: <<http://www.phpmsadmin.org/>> Acesso em: 06/06/2015.

código da ferramenta SQL Server, da Microsoft, e o PHPPgAdmin⁵⁵, cuja matriz de programação é o SGBD Postgre⁵⁶.

Assim como o MySQL, tanto no modo servidor quanto no modo cliente, o PHPMyAdmin também é capaz de criar sistemas de bancos de dados, incluir, excluir, consultar e editar tabelas e relacionamentos entre elas. O que torna esta ferramenta popular é a possibilidade de fazer tudo isso graficamente e com menos necessidade de se fazer as mesmas coisas de maneira procedural, isto é, escrever códigos, mas isto só se faz em casos específicos. Por exemplo, neste DB, foi necessário recorrer ao método procedural para relacionar uma lista (tabela de uma coluna só) com outra tabela maior, para evitar a redundância de dados.

Assim como o navegador *Firefox*, o PHPMyAdmin (FIGURA 6) é, também, uma ferramenta multi-plataforma. Mas, neste caso, não depende de versões específicas de sistema operacional para rodar, pois é uma ferramenta que roda em navegadores de Internet. Logo, é uma ferramenta portátil, já que roda só em navegadores de Internet, independentemente se está sendo utilizado no SO *Windows*, *Linux* ou *MacOS*.

⁵⁵ PHPPgAdmin: Administrador de banco de dados via página Web que fornece funcionalidades para o SGBD open-source Postgre. Disponível em: <<http://phppgadmin.sourceforge.net/doku.php>> Acesso em: 06/06/2015.

⁵⁶ Postgre: Sistema gerenciador de bancos de dados relacional orientado a objetos, desenvolvido em código aberto, concorrente do MySQL. Disponível em: <<http://www.postgresql.org/>> acesso em: 01/06/2015.

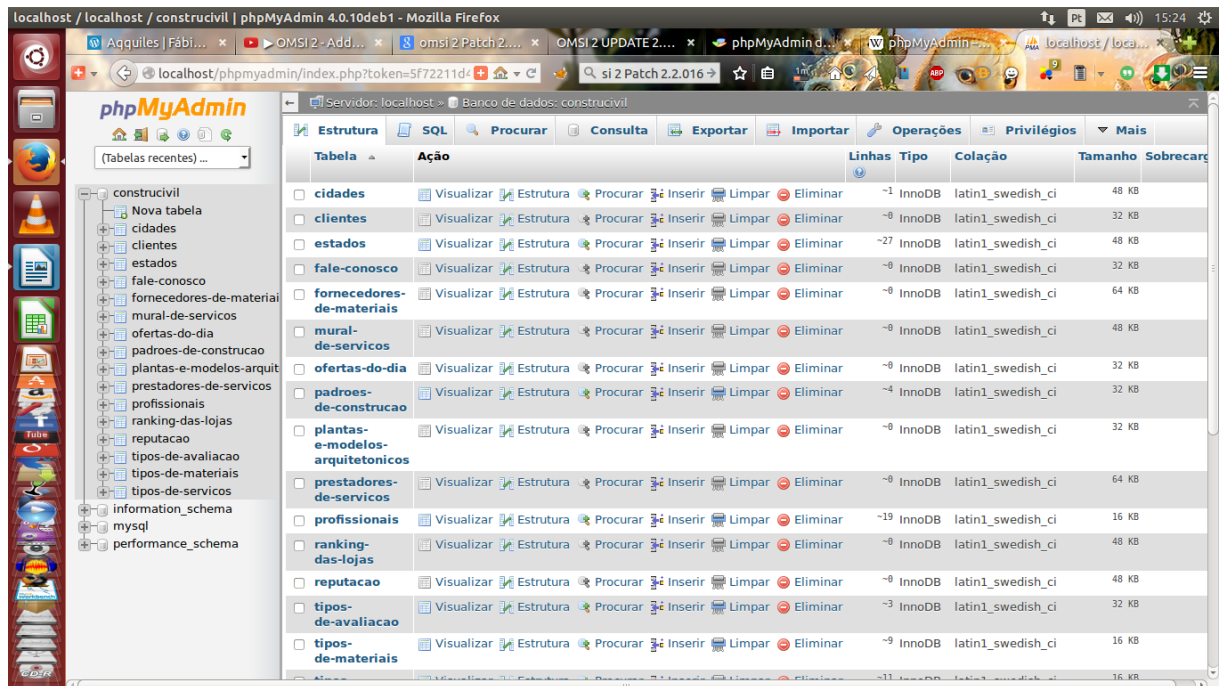


FIGURA 6: PHPMYADMIN.

FONTE: O Autor (2015).

4.3.7 MySQL Workbench

MySQL Workbench (FIGURA 7) é uma ferramenta GUI para o projeto de banco de dados, que integra o desenvolvimento de códigos SQL, formato padrão de *script* de banco de dados, gestão, modelagem de dados, criação e manutenção de banco de dados MySQL em um único ambiente sinérgico. Há versões para Linux e para *Windows*. É uma ferramenta de uso livre e gratuito, tem código-fonte aberto para desenvolvedores (ferramenta *open-source*).

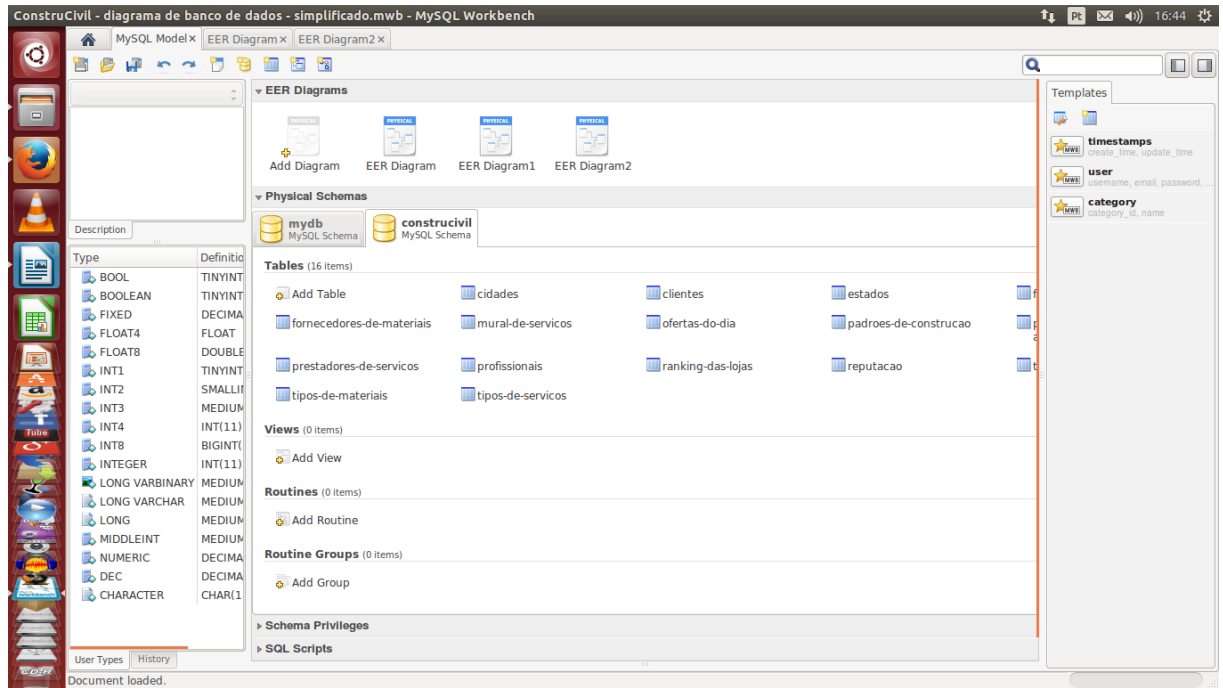


FIGURA 7: MYSQL WORKBENCH.

FONTE: O Autor (2015).

5 METODOLOGIA

A metodologia utilizada neste trabalho foi de pesquisa bibliográfica para criação de sistemas informatizados de gerenciamento e armazenamento de dados. Livros, artigos acadêmicos impressos, artigos em páginas *Web* e apresentações de *slides* foram os meios utilizados para moldar o presente conhecimento, além de Figuras de projetos bem-sucedidos de bancos de dados, o foco principal deste trabalho acadêmico.

Ainda envolveu depoimentos de pessoas que costumam “peregrinar” pela região em busca de produtos e serviços voltados à construção civil, como reformas, jardinagem, paisagismo e serviços que dependam de profissionais como arquiteto, paisagista, pedreiro e pintor.

Este trabalho acadêmico traz todo o conhecimento adquirido por meio de materiais citados anteriormente, além de trazer todos os detalhes do desenvolvimento do banco de dados (DB) “ConstruCivil” e as modificações ao longo do tempo de desenvolvimento até chegar à uma maturidade satisfatória. Para cada modificação realizada no banco de dados, uma nova versão, cuja nomenclatura será modificada conforme o ano, mês e dia em o trabalho foi sendo realizado.

6 DESENVOLVIMENTO DO SISTEMA

O Banco de Dados “ConstruCivil” é um banco de dados relacional, modelo comum utilizado em sistemas de informação que dependem de mecanismo de armazenamento de informações vitais à organização a ser estudada. O futuro portal “ConstrutorCivil”⁵⁷ receberá informações de produtos e serviços e até ofertas de trabalho e de produtos em promoção. Todas as informações a serem coletadas no portal irão alimentar o DB, e cada incremento nele receberá uma numeração diferente como uma forma de identificar esta linha como identificação única e intransferível.

6.1 MODELAGEM DO SISTEMA

Machado (2008, p. 16) explica que a modelagem de dados:

“É o estudo de informações existentes em um contexto sob observação para a construção de um modelo de representação e entendimento de tal contexto. A modelagem de dados minera as informações que representam um contexto, estruturando-as em um conjunto que denominamos modelo lógico de dados”.

Os dados foram modelados a partir do processo de levantamento de requisitos, conforme a FIGURA 8, desenhada em conjunto com os demais estudantes, durante as aulas de projetos de aprendizagem do curso de Informática e Cidadania, na UFPR Litoral. Na referida figura, é mostrado o procedimento de busca de produtos ou serviços por cidade. O primeiro procedimento é selecionar a cidade (sete municípios do litoral paranaense). Selecionada a cidade, um *script* será acionado de acordo com a cidade escolhida pelo usuário, e todas as informações referidas a ela estarão de prontidão para ele escolher entre produtos, serviços, lojas, contatos, entre outros.

⁵⁷ Construtor Civil. Portal de buscas de produtos e serviços na área de construção civil no Litoral do Paraná. Disponível em: <<http://www.construtorcivil.com.br>> Acesso em: 10/04/2015.

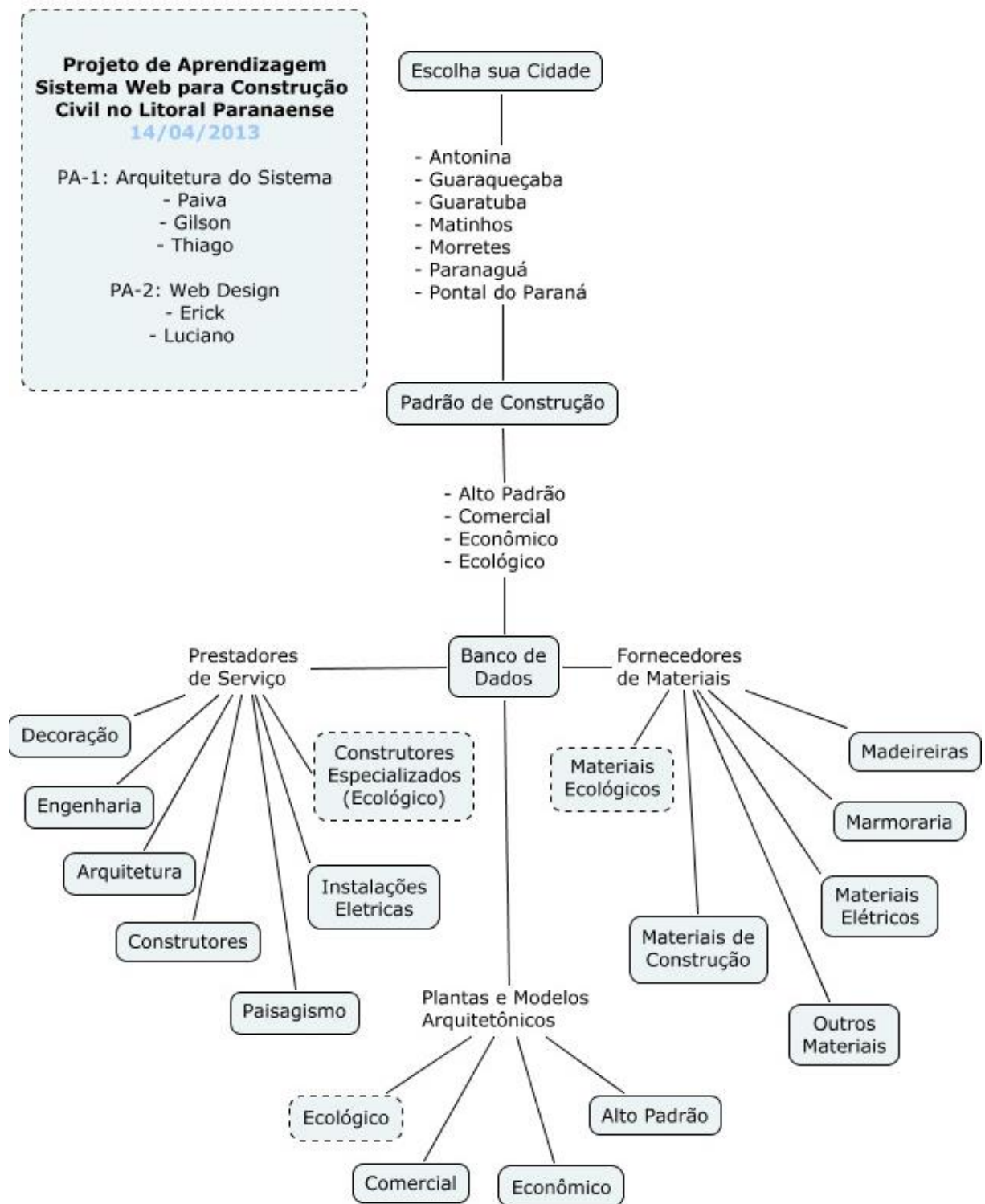


FIGURA 8: MAPA CONCEITUAL DO SISTEMA.

FONTE: O Autor (2015).

O banco de dados é dividido em quatro eixos temáticos:

- Plantas e modelos arquitetônicos;
- Fornecedores e lojas de materiais;
- Prestadores de serviços, profissionais liberais e autônomos;
- Mural de serviços.

6.1.1 Plantas e Modelos Arquitetônicos

O primeiro eixo temático trata como pesquisar empreendimentos por padrão de construção. São quatro padrões (Alto padrão, comercial, econômico e ecológico) diferentes para o usuário escolher. Trata-se de projetos de arquitetura, que são cadastrados por arquitetos ou por construtoras. Existe um campo especial para colocação de *links* para seu *site*, no qual existem mais detalhes do projeto de arquitetura ou empreendimento.

A FIGURA 9 representa o modelo entidade-relacionamento entre os padrões de construção e as plantas e modelos arquitetônicos.



FIGURA 9: MODELO ENTIDADE-RELACIONAMENTO ENTRE OS PADRÕES DE CONSTRUÇÃO E AS PLANTAS E MODELOS ARQUITETÔNICOS.

FONTE: O Autor (2015).

6.1.2 Fornecedores e Lojas de Materiais

O segundo eixo temático trata dos fornecedores e lojas de materiais. Eles são divididos por categorias (materiais ecológicos, materiais de construção, materiais elétricos, marmoraria, madeiras e outros materiais). Cada fornecedor ou loja deverá estar localizada em uma das sete cidades do litoral paranaense, ou em mais de uma em caso de haver filiais, com endereço completo, telefone e, se tiver, fax, endereço eletrônico e *e-mail*. Os usuários poderão, também, avaliar a qualidade dos serviços e produtos oferecidos pelas lojas. Mais informações nos subitens do item 6.2.1, que tratam das versões do banco de dados.

A FIGURA 10 representa o modelo entidade-relacionamento entre os fornecedores de materiais, os clientes e a cidade onde cada um se situa. Sabe-se que pode não haver fornecedores de materiais em uma determinada cidade, mas um fornecedor tem que estar instalado em uma cidade, e o cliente deve estar situado em uma cidade. Cidade pode ter um ou mais clientes e pode ter vários fornecedores ou pode não ter nenhum.

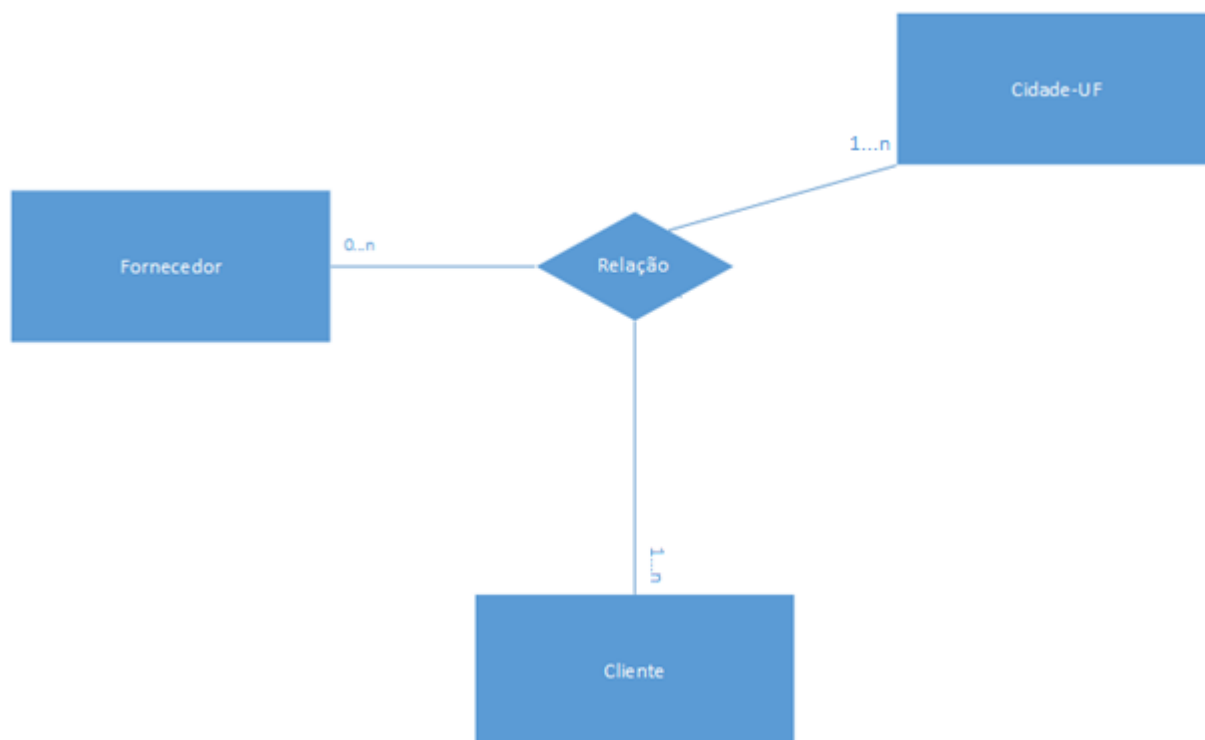


FIGURA 10: MODELO ENTIDADE-RELACIONAMENTO ENTRE A CIDADE E OS CLIENTES E OS FORNECEDORES DE MATERIAIS.

FONTE: O Autor (2015).

6.1.3 Prestadores de Serviços, Profissionais Liberais e Autônomos

O terceiro eixo temático trata dos prestadores de serviços. Estão cadastrados os tipos de serviços: arquitetura, construção, construção especializada, decoração, encanamento, engenharia, instalações elétricas, paisagismo, reparos, terraplanagem e outros não informados. Também está atrelado a este eixo os profissionais autônomos e liberais: arquiteto, azulejista, carpinteiro, decorador, eletricitista, encanador, engenheiro civil, engenheiro eletricitista, jardineiro, marceneiro, mestre de obras, paisagista, pedreiro (inclui servente de pedreiro também), pintor, projetista, roçadeiro, serralheiro e vidraceiro. Os usuários também podem avaliar o trabalho de cada um desses profissionais, graças às tabelas “Tipos de avaliação” e “reputação”, que se relacionam entre si para alimentar as tabelas “Avaliação”, no caso dos profissionais autônomos e liberais, e “Ranking das lojas”, no caso dos fornecedores de materiais.

Quanto à avaliação dos profissionais e lojas, a FIGURA 11 representa como as tabelas que fazem parte deste caso de uso se comunicam e como elas são alimentadas.

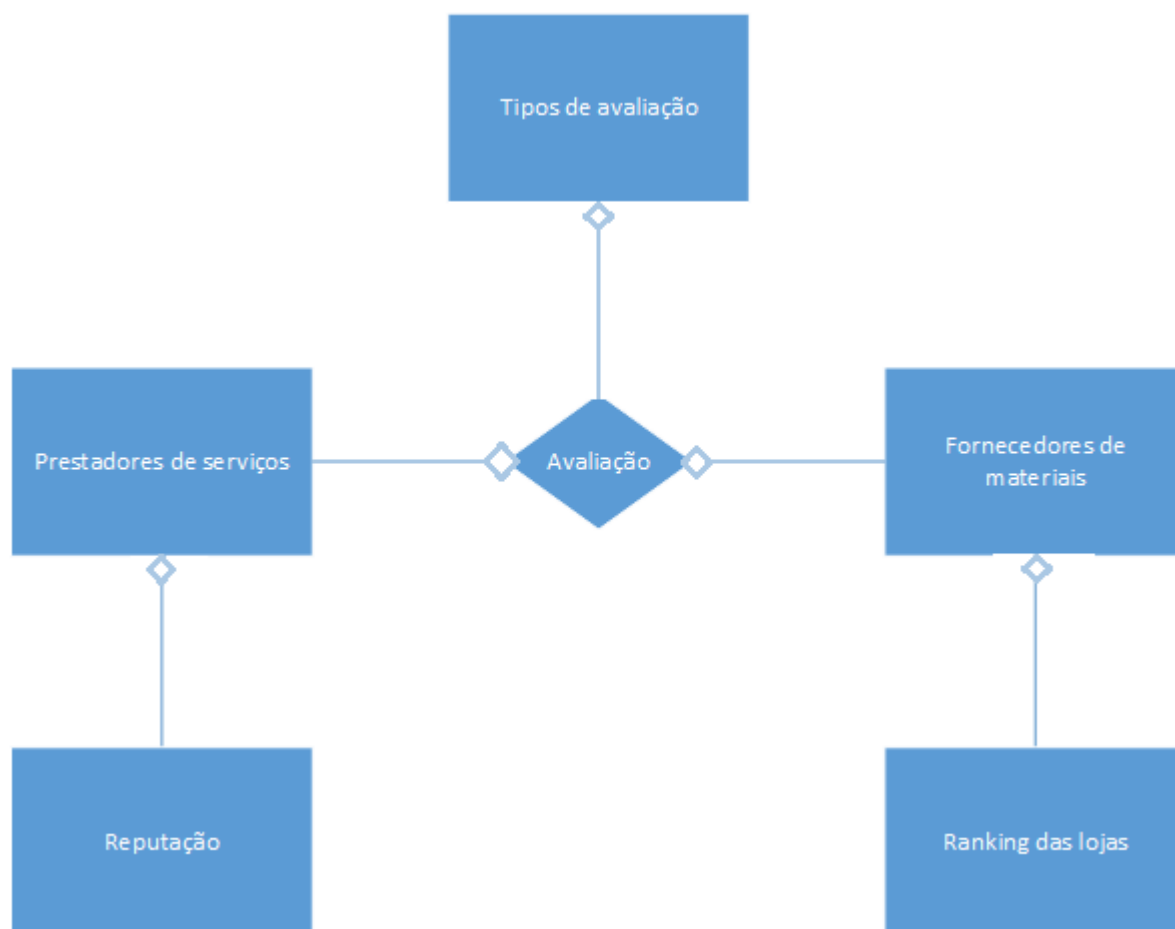


FIGURA 11: MODELO ENTIDADE-RELACIONAMENTO – SUBSISTEMA AVALIAÇÃO E REPUTAÇÃO.

FONTE: O autor (2015).

Na FIGURA 11, é retratado que a tabela “Tipos de avaliação”, que contém as avaliações cadastradas (excelente, ótimo e bom), alimenta as tabelas “Prestadores de serviços”, cujas avaliações serão somadas de acordo com o tipo de avaliação e suas informações alimentarão a tabela “Reputação”, e a tabela “Avaliação” alimentará também a tabela “Fornecedores de materiais”, cujas avaliações serão somadas de acordo com o tipo de avaliação, assim como a tabela que representa os prestadores de serviços. A soma de cada tipo de avaliação terá sua soma correspondente a cada fornecedor ou loja representada na tabela “Ranking das lojas”.

6.1.4 Mural de Serviços

O quarto eixo temático trata sobre a oferta de produtos e serviços. Este eixo é dividido em dois subsistemas: o mural de serviços e a oferta de produtos em promoção.

Quanto ao mural de serviços, os clientes cadastrarão ofertas de serviço que requisitam na tabela “Mural de serviços”. E os profissionais cadastrados saberão que há ofertas de trabalho. O cliente que vai requisitar um serviço escolherá o tipo de serviço e, conforme o tipo de serviço selecionado, o profissional saberá da oferta de trabalho de acordo com a sua área. Por exemplo: um cliente cadastra uma oferta de trabalho e seleciona o tipo de serviço chamado “Reparos”. O prestador de serviço que identificar-se como quem faz reparos saberá ao visitar regularmente o *site* “construtorcivil.com.br” que há uma oferta de trabalho para ele.

Já a oferta de produtos em promoção se dará por parte dos fornecedores de materiais. A tabela “Ofertas do dia”, que foi criada a partir da versão 20140407 conforme o item 6.2.1.2, é destinada à oferta de produtos que as lojas e fornecedores em geral colocam à disposição do público em promoção ou destaque. O fornecedor deverá estar cadastrado no *site* “construtorcivil.com.br” para poder realizar o cadastro do produto. Ele informará o nome do produto, o preço, data e hora do início da promoção e a data e hora do final da promoção. Quando a promoção terminar, a programação do *site* ficará encarregada de excluir a promoção automaticamente.

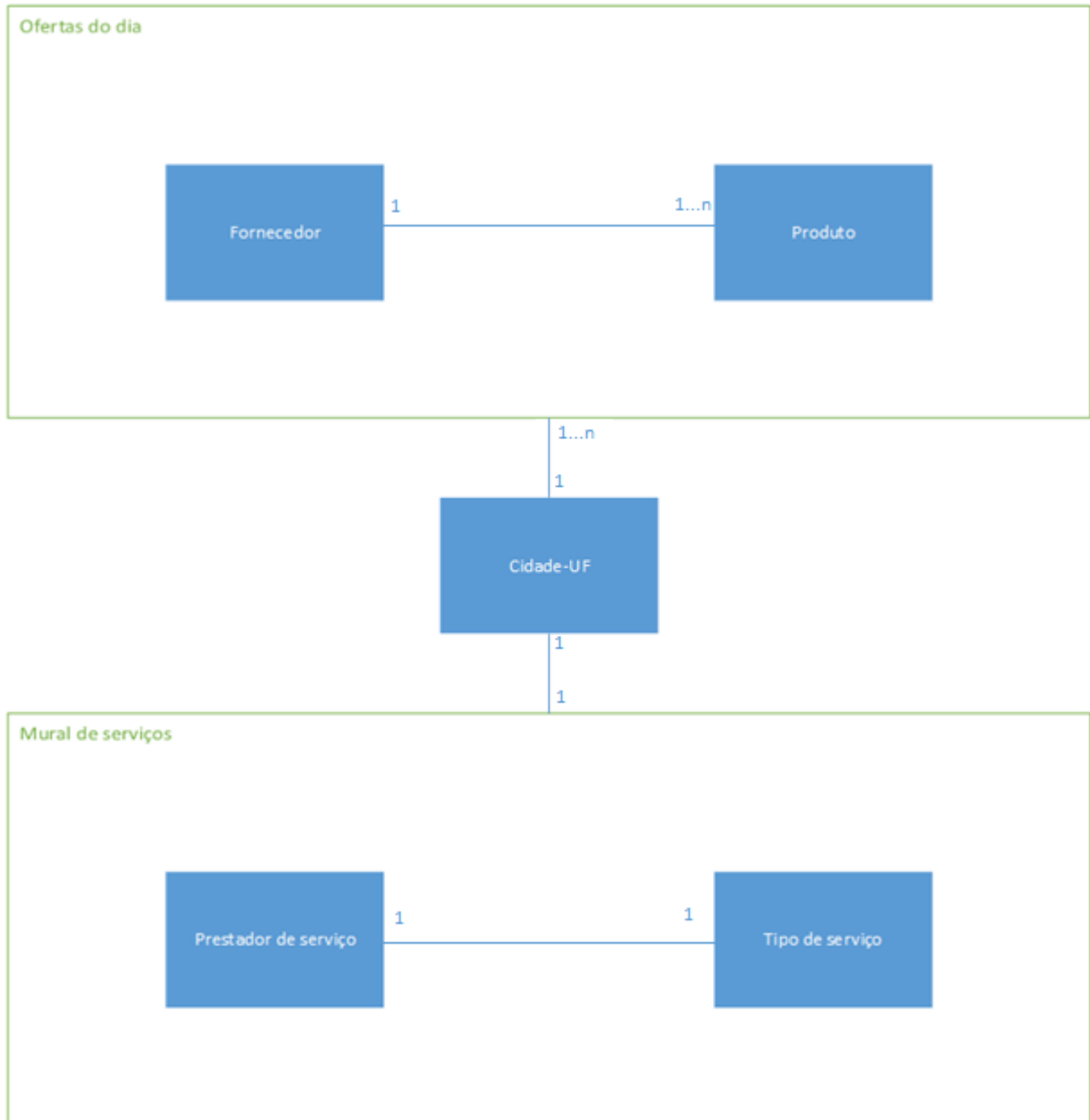


FIGURA 12: MODELO ENTIDADE-RELACIONAMENTO DO SUBSISTEMA OFERTA DE PRODUTOS E SERVIÇOS.
FONTE: O Autor (2015).

6.2 VERSÕES DO BANCO DE DADOS E SUAS MODIFICAÇÕES

O banco de dados “ConstruCivil” teve diversas versões conforme modificações são realizadas ao longo da jornada, desde a concepção até o

aperfeiçoamento dele. O processo de modelagem de dados é importante para separar os objetos, de acordo com a função que desempenha. O objetivo é padronizar informações consideradas chaves e que não devem ser digitadas de maneiras diferentes. Do contrário, haverá dificuldades para o desenvolvimento de *scripts* para pesquisar coisas por critério definido.

6.2.1 Versões

Cada versão do banco de dados recebe um nome para distingui-la das demais. A primeira versão foi criada no dia 3 de outubro de 2013. E o nome desta primeira versão recebeu a numeração, de acordo com a data de criação, e contém oito dígitos. Os quatro primeiros dígitos correspondem ao ano que, neste caso, é 2013. Os dois dígitos seguintes correspondem ao mês que, neste caso, é outubro. E os dois dígitos restantes correspondem ao dia 3. O resultado da nomenclatura adotada neste trabalho acadêmico será 20131003. E cada nova versão que é criada, o número será dado conforme a nomenclatura adotada. Por exemplo, se é criada uma nova versão no dia 3 de maio de 2015, a numeração para esta versão será 20150503, e assim, sucessivamente.

Na primeira versão do DB “ConstruCivil”, as tabelas não tinham relacionamento entre si. Também não eram normalizadas. Toda a evolução do DB é vista a seguir.

6.2.1.1 Versão 20131003

Esta foi a primeira versão do projeto, e foi baseado no mapa conceitual, descrito na FIGURA 12. Nesta versão, o banco de dados era composto por 8 tabelas. Todas têm chave, ou primária ou única.

A partir deste arquivo, todas as posteriores versões do banco de dados “ConstruCivil” foram desenvolvidas.

A primeira tabela criada foi “Cidades”. A chave primária⁵⁸ é a coluna ‘Código da cidade’. Chave primária serve para dar uma identificação única para uma tupla⁵⁹ (linha de registro). Battisti (2015, pg. 1) explica que chave primária “identifica de Maneira Única cada registro de uma tabela, isto é, de posse do valor da Chave Primária somente” se localizará “um registro com aquele valor no campo Chave Primária”. E também a coluna ‘Cidade-UF’ e ‘Estado’. Quando esta tabela foi montada, foi pensado em usar a coluna ‘Estado’ como sigla da unidade federativa correspondente (duas letras, como em SP, PR, RS e SC). Este campo fora usado apenas para fins de consulta de cidades por estado. Ainda não fora encontrada uma maneira de listar cidades por estado sem a coluna ‘Estado’, por mais que se inserisse no script algo como

```
select `Cidade-UF` from `ConstruCivil`.`Cidades` where `UF` ==
`PR`
```

Com este exemplo, o SGBD não entenderia e retornaria uma mensagem de erro. A tabela “Cidades” alimentará outras tabelas, como será relatado mais adiante.

Em seguida, foi montada a tabela “Clientes”. Sua chave primária é a coluna ‘Código do Cliente’, código aleatório para identificar um único cliente. Contém 6 dígitos, e usa a sintaxe Ano-mês-nº sequencial, contém 6 dígitos e é auto incrementável. Na sequência, foram incluídas as colunas ‘Nome do cliente’, ‘Endereço’, ‘CEP’, ‘E-mail’ (se não tiver, não é necessário informar) e ‘Telefone’ (também não é necessário informar). Logo, a coluna ‘Cidade-UF’ obtém dentro de uma lista, todas as cidades cadastradas na tabela “Cidades”, na coluna ‘Cidade-UF’ naquela mesma tabela.

Na sequência, foi montada a tabela “Fornecedores de materiais”. A chave primária dela foi intitulada ‘Código do cliente’, contém 5 dígitos e é auto incrementável. Chama-se assim porque cada registro na tabela terá uma numeração diferente. Ao fazer um novo cadastro, o sistema gerenciador de banco de dados (SGBD) somará 1 no número do código. Por exemplo, o primeiro registro (tupla)

⁵⁸ Chave primária: Campo que identifica uma linha de uma tabela de dados. Disponível em: <<http://www.1keydata.com/pt/sql/sql-chave-primaria.php>> Acesso em: 06/06/2015.

⁵⁹ Tupla: linha de uma tabela de dados, formada por uma combinação de colunas para representar um objeto em uma única estrutura. Disponível em: <<http://www.facom.ufu.br/~madrana/PF/pratica01.pdf>> Acesso em: 06/06/2015.

nesta tabela terá o código 13001. Depois, os registros seguintes terão este código somando 1. Sendo assim, a primeira tupla terá o registro 13001, e o segundo terá o registro 13002, e depois 13003, e depois 13004, e assim por diante. Foi incluída a coluna 'Nome do fornecedor', que serve para cadastrar o nome ou razão social da empresa fornecedora, podendo ser uma loja de materiais de construção, uma marmoraria, uma floricultura, uma loja de decoração, entre outros. A coluna 'Tipo de material' é do tipo índice, ou seja, recebe informações de uma coluna chave, do mesmo tipo, de outra tabela. No caso desta coluna, ela recebe informações previamente cadastradas de uma coluna chamada 'Tipo de material' da tabela "Tipos de Materiais". Há ainda as colunas 'Endereço', 'CEP' e 'Telefone'.

Todas as colunas citadas até o momento são obrigatórias que sejam preenchidas. As colunas 'Fax' e 'Email' não são obrigatórias que sejam preenchidas, considerando que nem todos possuem número de fax ou endereço de correio eletrônico (e-mail), e assim, fica a critério do usuário registrar ou não uma ou as duas informações no formulário de cadastro, que será disponibilizada no site 'ConstrutorCivil'. A coluna 'Cidade-UF' é do tipo índice, assim como a coluna 'Tipo de material'. No caso da coluna 'Cidade-UF', ela é alimentada pela coluna 'Cidade-UF' da tabela "Cidades". Então, quando um usuário se cadastrar, no momento em que for incluir a cidade do seu empreendimento, ao clicar no campo 'Cidade-UF', uma lista de cidades aparecerá, com todas as informações previamente cadastradas no banco de dados.

As tabelas "Padrões de Construção" e "Plantas e modelos arquitetônicos". Elas não têm relacionamento algum com as outras tabelas do banco de dados, só têm relacionamento entre si. Esta é uma peculiaridade, se levar em conta um padrão de montagem de banco de dados relacional. Na tabela "Padrões de Construção", existe um identificador (chave primária) chamado 'código PC', de dois dígitos, e a coluna 'Padrão de construção', cujas modalidades estão classificadas como: ecológico, econômico, comercial e alto padrão, variando, quanto ao custo, do mais baixo ao mais elevado, dependendo do poder aquisitivo do usuário. Em contrapartida, a tabela "Plantas e projetos arquitetônicos", será discutida nas versões posteriores, pois ainda não havia sido cadastrada nesta versão do banco de dados.

Em seguida, foi criada a tabela “Prestadores de serviços”. A chave primária é a coluna ‘Código do prestador de serviço’, tem 5 dígitos e é auto incrementável. Foi incluída a coluna ‘Nome do ou razão social’, que serve para cadastrar o nome do profissional ou de uma empresa, podendo ser uma pessoa física ou jurídica, profissional autônomo ou profissional liberal. A coluna ‘Profissão, cargo ou tipo de serviço’, por ser uma chave tipo índice, será alimentada pela coluna ‘Profissão, cargo ou tipo de serviço’ da tabela “Profissionais”, onde as informações estarão cadastradas primeiramente. Significa que, quando o usuário cadastrar um novo prestador de serviços, ao clicar sobre o campo ‘Profissão, cargo ou tipo de serviço’, surgirá uma lista com todas as opções disponíveis. As colunas ‘Endereço’, ‘Telefone’ e ‘CEP’, assim como todas as colunas anteriores nesta tabela, são também de preenchimento obrigatório, como é também a coluna ‘Cidade-UF’ (coluna índice), que é alimentada pela coluna do mesmo nome, na tabela “Cidades”. O único campo em que o preenchimento não é obrigatório se chama ‘Fax’, pois nem todos os prestadores de serviços dispõem do mesmo.

A tabela “Profissionais” é, na verdade, uma lista, matriz unidimensional, ou seja, só existe uma coluna, um único domínio de tabela, e se chama ‘Profissão, cargo ou tipo de serviço’. Ela alimenta outras colunas de outras tabelas, que serão discutidas durante esta documentação. Não se descarta, também, a possibilidade de incluir novas tabelas que, por ventura, venham a depender, em algum momento, de informações a serem incluídas nesta tabela. Esta tabela não contém numeração como chave primária. A chave identificadora é a própria coluna ‘Profissão, cargo ou tipo de serviço’, cujos registros serão únicos. Sendo assim, acreditou-se que não haverá necessidade de incluir uma coluna para identificar uma profissão ou cargo, já que não serão escritos mais de uma vez.

Outras tabelas unidimensionais criadas neste banco de dados são as tabelas “Tipos de materiais” (o identificador é a coluna ‘Tipo de material’) e “Tipos de serviços” (o identificador é a coluna ‘Tipo de serviço’). Essas duas tabelas unidimensionais alimentarão colunas homônimas em outras tabelas. Mais detalhes no decorrer desta documentação.

6.2.1.2 Versão 20140407

Foi criada no dia 7 de abril de 2014. Foram realizadas alterações na tabela “Fornecedores de materiais”. Nela, foi adicionada a coluna “site”. Preencher esta coluna não é obrigatória para completar o cadastro, assim como o campo “FAX”, considerando que nem todas as lojas possuem essas ferramentas de comunicação. No dia da criação desta versão, em 7 de julho de 2014, a chave primária (código do fornecedor) contém 5 dígitos e recebeu um padrão de auto incremento, contando a partir do número 14001. Cada registro na tabela terá uma numeração diferente. Ao fazer um novo cadastro, o sistema gerenciador de banco de dados (SGBD) somará 1 no número do código. Sendo assim, sabendo que, no primeiro registro da tabela, o código do fornecedor foi 14001, no cadastro seguinte, caso o administrador do banco não dê uma numeração diferente, o SGBD dará como novo registro o código 14002, e depois 14003, e depois 14004, e assim por diante. Mesmo assim, o administrador pode dar uma numeração diferente quando achar conveniente. Por exemplo, a partir do ano 2015, ele pode preferir colocar como código do fornecedor o número 15001, sendo que 15 representa o ano 2015, e os três dígitos seguintes representam a ordem sequencial do cadastro.

Nesta versão, foi incluída a tabela “Mural de serviços”, onde serão cadastradas as ofertas de trabalho para profissionais liberais e prestadores de serviços. Aqui ainda, clientes ou empresas podem procurar por serviços especializados. E ofertá-los também. Todos os campos devem ser preenchidos. A chave primária é o campo “ID” e contém 10 dígitos e é auto incrementável. O campo “Tipo de Serviço” está relacionado com o campo do mesmo nome, da tabela “Tipos de serviços”, de onde esta tabela vai obter a informação descrita como o tipo de serviço que o requisitante está demandando, como ‘reforma’, ‘encanamento’, ‘reparos’, ‘construção’, etc. O campo “Cidade do projeto” está relacionado com a tabela “Cidades”, de onde esta tabela vai obter a informação descrita como a cidade e a unidade federativa, onde o serviço requisitado será empreendido.

Foi criada neste banco de dados também a tabela “Ofertas do dia”. Aqui, os fornecedores podem lançar ofertas em promoção de um determinado produto em sua loja. A chave primária desta tabela é o campo “Código da oferta”, que contém 10 dígitos e é auto incrementável. O administrador do banco de dados configurou para

que este código seja gerado automaticamente, a partir da 140405002, usando a sintaxe ‘ano-mês-dia-nº sequencial’, apenas por conveniência. Neste caso, a data do primeiro cadastro nesta tabela ocorreu no dia 5 de abril de 2014 e foi o número dois deste mesmo dia. O campo “Nome da loja” está relacionado com a coluna “Nome do Fornecedor” da tabela “Fornecedores de Materiais”, onde esta tabela vai obter a relação completa de todos os fornecedores cadastrados. Um dado importante é que apenas os fornecedores cadastrados no *site* podem cadastrar seus produtos para lançar suas ofertas promocionais no portal.

Cada promoção tem uma data inicial e uma data final. Então, nos campos “Início da promoção” e “Término da promoção”, a sintaxe, definida com data e hora, para as duas colunas é ‘ano-mês-dia hora:minuto:segundo’, como em “2013-11-23 18:00:00”. Esta é a sintaxe que o SGBD utiliza.

Nesta versão, foram incluídas as tabelas “Ranking das lojas”, “Reputação” e “Tipos de avaliação”. Essas três tabelas estão relacionadas entre si, a partir da tabela “Tipos de avaliação” que serve às outras duas. Esta tabela oferta ao público a possibilidade de medir o nível de satisfação dos usuários. Na tabela “Reputação”, a coluna ‘avaliação’ está relacionada com a coluna ‘avaliação’ da tabela “Tipos de avaliação”, de onde obtém as informações previamente cadastradas. Ainda em “Reputação”, o usuário digita o nome da empresa ou de algum profissional liberal na coluna ‘Nome ou razão social’ e seleciona o tipo de avaliação na coluna ‘Avaliação’ (Excelente, ótimo ou bom). Já a tabela “Ranking das lojas” obtém informações das tabelas “Tipos de avaliação”, por meio da coluna ‘Avaliação’, e “Fornecedores de materiais”, por meio da coluna ‘Nome ou razão social’, que alimentará a coluna ‘Loja’ na tabela “Ranking das lojas”. Aqui, o usuário seleciona o nome da loja já cadastrada no site e seleciona a avaliação, conforme seu nível de satisfação quanto ao atendimento.

Existem tabelas com uma coluna só, que na verdade são denominados vetores, ou ainda, listas de valores. É o caso da tabela “Plantas e Modelos Arquitetônicos”, que foi incluída, pela primeira vez no DB “ConstruCivil” nesta versão. Ele serve como ponto de referência para a divulgação de plantas e modelos de projetos de arquitetura e construção. Arquitetos cadastrados no sistema poderão divulgar seus projetos de lançamento de seus empreendimentos. A chave primária é

o campo “ID”, que servirá como identificação única para cada linha a ser incluída na tabela, ainda que as informações nas colunas estejam parecidas ou iguais, mas o ID é único para cada linha. É um campo numérico auto incrementável. A coluna “Nome do projeto” refere-se ao nome oficial, razão ou fantasia do empreendimento a ser divulgado. O campo “Padrão de construção” está relacionado com a coluna “Padrão de construção” da tabela “Padrões de construção”, que já foi alimentada e tem 4 linhas, cada uma representando uma categoria de acordo com o custo do projeto (alto padrão, comercial, ecológico e econômico). O campo “Descrição” é uma caixa de texto livre. Refere-se à descrição da planta ou modelo arquitetônico. Existe, ainda, a coluna “Data-do-projeto”, que se refere à data de lançamento do empreendimento. Encerrando, o campo “Link”, que é o vínculo de ligação entre o site “ConstrutorCivil” e o do arquiteto ou qualquer empreendedor responsável pelo modelo arquitetônico, caso exista tal site dedicado a ele. O arquiteto pode inserir o endereço eletrônico do que pode ser uma figura ilustrativa ou um arquivo de texto em PDF, para fins de Figura do empreendimento.

A FIGURA 13 a seguir mostra como está organizado o sistema de banco de dados, com suas tabelas e relacionamentos. Serve para demonstrar a dinâmica do desenvolvimento do banco de dados.

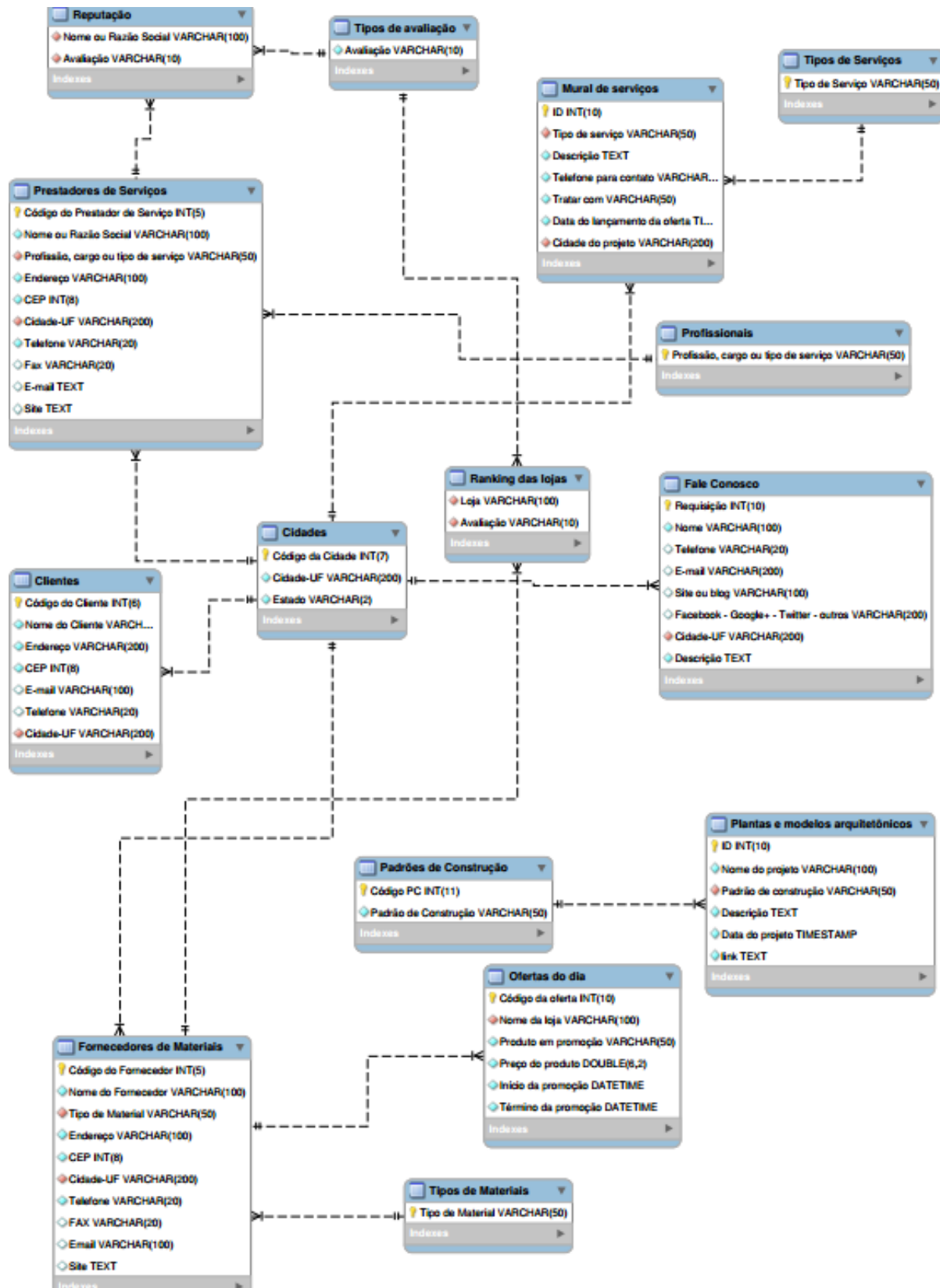


FIGURA 13: DIAGRAMA DE BANCO DE DADOS VERSÃO 20140407, DESENVOLVIDO COM A FERRAMENTA MYSQL WORKBENCH.

FONTE: O Autor (2015).

6.2.1.3 Versão 20140516

Foi criada no dia 6 de maio de 2014. Foi incluída, neste banco de dados, a tabela “Fale conosco”. A chave primária é a coluna ‘Requisição’, que contém 10 dígitos e é auto incrementável. Os únicos campos onde o preenchimento é obrigatório são ‘Nome’, ‘Cidade-UF’ e ‘Descrição’, que é uma coluna de tipo TEXTO, que permite que o usuário redija um texto de tamanho indefinido. A coluna ‘Cidade-UF’ é do tipo ÍNDICE e é alimentada pela coluna homônima, na tabela “Cidades”. Os outros campos, o usuário não tem a obrigação de preencher, ou seja, o preenchimento é facultativo, dependendo do interesse dele. São eles: ‘Telefone’, ‘Email’, ‘Site ou blog’ e ‘Facebook - Google+ - Twitter - outros’.

Na tabela “Mural de serviços”, houve uma alteração. A chave primária, identificada como ‘ID’ ganhou mais dígitos, tendo, ao todo, 10. ‘ID’ é um código aleatório gerado automaticamente pelo sistema, e logo, é auto incrementável. A sintaxe aplicada foi definida como ‘ano-mês-dia-nº sequencial’. Exemplo: 1404070001.

6.2.1.4 Versão 20140716

Visualmente, não há alterações significativas, se comparar esta versão com a anterior. Só foi substituída a ferramenta de desenvolvimento do banco de dados, que rodava no servidor “WAMP”⁶⁰. Todo o banco de dados foi transferido para o ambiente “XAMPP”⁶¹, mas que também roda o servidor MySQL, já com outra versão. Apenas foi renumerada para outra versão, apenas para fins de revisão de consistência do DB.

⁶⁰ WAMP: Pacote de aplicativos de desenvolvimento que envolve o servidor Web Apache, o SGBD MySQL e as linguagens de programação PHP, Perl e Python. Este pacote foi desenvolvido exclusivamente para o Sistema Operacional Windows. Disponível em: <<http://www.wampserver.com/en/>> acesso em: 01/06/2015.

⁶¹ XAMPP: Pacote de softwares livres, voltado para o desenvolvimento e de plataforma independente. No pacote, estão disponíveis os mesmos programas do Wamp. Disponível em: <http://www.apachefriends.org/pt_br/index.html> acesso em: 01/06/2015.

A partir daqui, porém, começaram a aparecer problemas com o DB. Havia problemas de consistência do banco e foi descoberto que o interpretador de códigos do MySQL não reconhece caracteres latinos como “ç” e letras com acentuações. Esta versão foi a primeira a ser testada fora de um servidor local e foi para um servidor de Internet. Assim se pode tomar conhecimento deste problema.

6.2.1.5 Versão 20140901

Nesta versão, os problemas relacionados à leitura e interpretação de código foram corrigidos, mas nem todos os problemas foram solucionados. Os nomes das tabelas e os nomes das colunas possuíam espaçamentos, e o interpretador de código do MySQL no servidor de Internet não podia reconhecer tal caractere, o que ocasionava erros de leitura.

6.2.1.6 Versão 20140930

Nesta nova versão, foram suprimidos os caracteres não-imprimíveis (espaços) nas variáveis (colunas) e nos nomes das tabelas. Esses espaços foram substituídos por traços. A acentuação das palavras também teve que ser suprimida. Por exemplo, onde estava escrito “Padrões de construção”, ficou “Padroes-de-construcao”. Depois de alguns testes extras no servidor de Internet, os problemas relacionados à leitura do código foram solucionados.

6.2.1.7 Versão 20141017

A partir desta versão, foi preferido não mais preencher colunas em tabelas que herdavam informações de outras tabelas, por falta de compreensão a respeito do problema. No caso do exemplo citado na versão anterior, a tabela “Fale-conosco” (anteriormente “Fale conosco”) possui o campo “Cidade-UF” obtém informações da coluna homônima da tabela “Cidades”. É provável que tal fato tenha causado conflito de informações de tal forma que prejudicou toda a confiabilidade do banco de dados.

Até mesmo essa mesma tabela ficou sem preencher os códigos, os nomes das cidades e suas respectivas unidades federativas. Sendo assim, foi decidido que as colunas deverão ser preenchidas após a instalação do DB na máquina, para fins de conveniência, a fim de manter a confiabilidade no código.

6.2.1.8 Versão 20141108

A partir desta versão, foi pensado em como melhorar o código quanto à tabela “Cidades”. Como de praxe, repetia-se com frequência o seguinte: na tabela “Cidades”, existe os campos “Codigo-da-cidade”, “Cidade-UF” e “UF”. No ato de cadastrar a cidade de Paranaguá, por exemplo, o DBA⁶² digita da seguinte forma: 4118204, 'Paranagua-PR', 'PR'.

Isto ocorreu até esta versão, pois ainda não havia sido encontrada uma maneira de desvincular a informação da unidade federativa do nome da cidade, quer dizer, criar um campo para a unidade federativa, separado da coluna destinada a preencher o nome da cidade, que deve herdar as informações de outra tabela, de forma que corresponda com a cidade em questão.

Assim, foi criada uma nova tabela, chamada “Estados”, com duas colunas, a chave primária “codigo_estado” e “UF”. As informações inseridas são baseadas em planilha de dados do IBGE (Instituto Brasileiro de Geografia e Estatística). No caso do estado do Paraná, por exemplo, o valor para o código do estado é 41 e a sigla é PR. Desta maneira, foi cadastrada essa informação na nova tabela.

No entanto, um novo problema surgiu. Ao referenciar a coluna “UF” da tabela “Cidades” com a coluna homônima da tabela “Estados”, houve redundância de dados. Por exemplo, na hora de cadastrar um cliente na tabela “Clientes”, foi necessário escolher a cidade onde ele reside. E na hora de escolher a cidade de Paranaguá, cadastrada no DB, apareceu duas vezes em sequências diferentes, conforme a FIGURA 14:

⁶² DBA: Database Administrator.

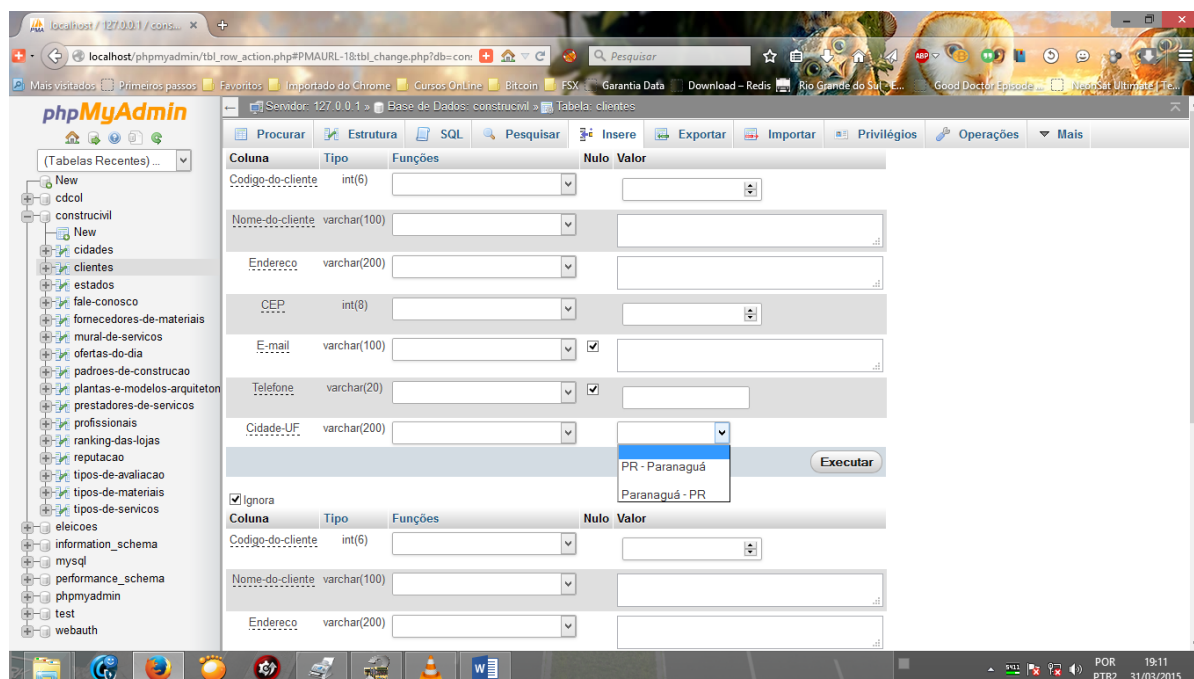


FIGURA 14: OCORRÊNCIA DE REDUNDÂNCIA DE DADOS.
FONTE: O Autor (2015).

Foi necessário, então, repensar na maneira de buscar uma lista das cidades pela sigla do estado. Uma nova versão do DB teve que ser desenvolvida.

6.2.1.9 Versão 20141110

Nesta versão, o problema, mostrado na FIGURA 14, relacionado à redundância de dados foi solucionado. Agora, não é mais mostrada a unidade federativa, uma vez percebida, que não é necessário listar cidades de outros estados, porque o cenário deste trabalho acadêmico é o Litoral Paranaense. É necessário, então, manter o foco nas cidades desta região. Quanto à tabela Estados, ela poderá ser utilizada, porém, ocultada ao público, para que os desenvolvedores tenham a exclusividade de trabalhar com a informação. O máximo que esta tabela servirá, será para listar as cidades de determinado estado para ocasiões futuras.

Nesta versão, no entanto, ainda há relacionamentos pendentes. É necessário relacionar a coluna “Avaliacao” da tabela “tipos-de-avaliacao” com a

coluna homônima da tabela “fornecedores-de-materiais”. Do ponto de vista da programação, sem tal solução, nem redundância de dados haverá, mas dará margem a usuários mal-intencionados a digitar termos inadequados, uma consequência da não-padronização de informações vitais ao negócio. O objetivo da solução é estabelecer uma relação entre o fornecedor de materiais e a clientela, de modo a transmitir confiança mútua. Quanto melhor a avaliação do fornecedor ou da loja ou algo do gênero, maiores as possibilidades de o cliente procurá-lo para fazer negócio. Portais eletrônicos como OLX⁶³, Buscapé⁶⁴, Zap⁶⁵, Mercado Livre⁶⁶ e Zoom⁶⁷ também adotam essa prática. Eles, porém, utilizam notas numéricas que vão de 5 a 10.

6.2.1.10 Versão 20150405

Esta é a versão mais recente do DB “ConstruCivil”. A modificação mais relevante para esta versão está relacionada ao relacionamento entre tabelas ligadas às tabelas “Cidades” e “Estados”. Conforme foi relatado na relação anterior, o problema de redundância de dados entre essas duas tabelas foi corrigido. Porém, a tabela “Estados” foi ocultada do público e servirá apenas para auxiliar na pesquisa de cidades conforme unidade federativa. Mas, como apenas 7 municípios, os que compõem o litoral do Paraná, serão listados no portal eletrônico “Construtor Civil”, não há necessidade de criar tal filtro por estado da federação. Em todas as tabelas que tiverem relação com a tabela “Cidades”, toda coluna que fora identificada como “Cidade-UF” teve o nome modificado para “Cidade”, em razão das modificações relatada há pouco.

Eis a seguir o diagrama de banco de dados, com todas as tabelas implementadas até aqui. Conforme a FIGURA 15, só serão mostrados os nomes das

⁶³ OLX: Empresa argentina que atua na área de anúncios classificados na Internet. Está presente em 118 países. Disponível em: <<http://www.olx.com.br/>> acesso em: 01/06/2015.

⁶⁴ Buscapé: Filial brasileira do grupo sul-africano Naspers. É especializada em pesquisa de preços de produtos. Disponível em: <<http://www.buscapede.com.br/>> acesso em: 01/06/2015.

⁶⁵ Zap: Empresa brasileira que atua no mercado imobiliário como veículo de publicidade. Disponível em: <<http://www.zapimoveis.com.br/>> acesso em: 01/06/2015.

⁶⁶ Mercado Livre: Empresa de comércio eletrônico via Internet fundada na Argentina. Disponível em: <<http://www.mercadolivre.com.br/>> acesso em: 01/06/2015.

⁶⁷ Zoom: Empresa brasileira de comércio eletrônico e pesquisa de preços. Disponível em: <<http://www.zoom.com.br/>> acesso em: 01/06/2015.

tabelas e seus relacionamentos, e não com todas as colunas, para poupar a atenção do leitor, até porque apenas uma coluna, ou duas colunas no máximo, farão ligação com outras tabelas. As colunas foram ocultadas somente para demonstração de relacionamento entre tabelas.

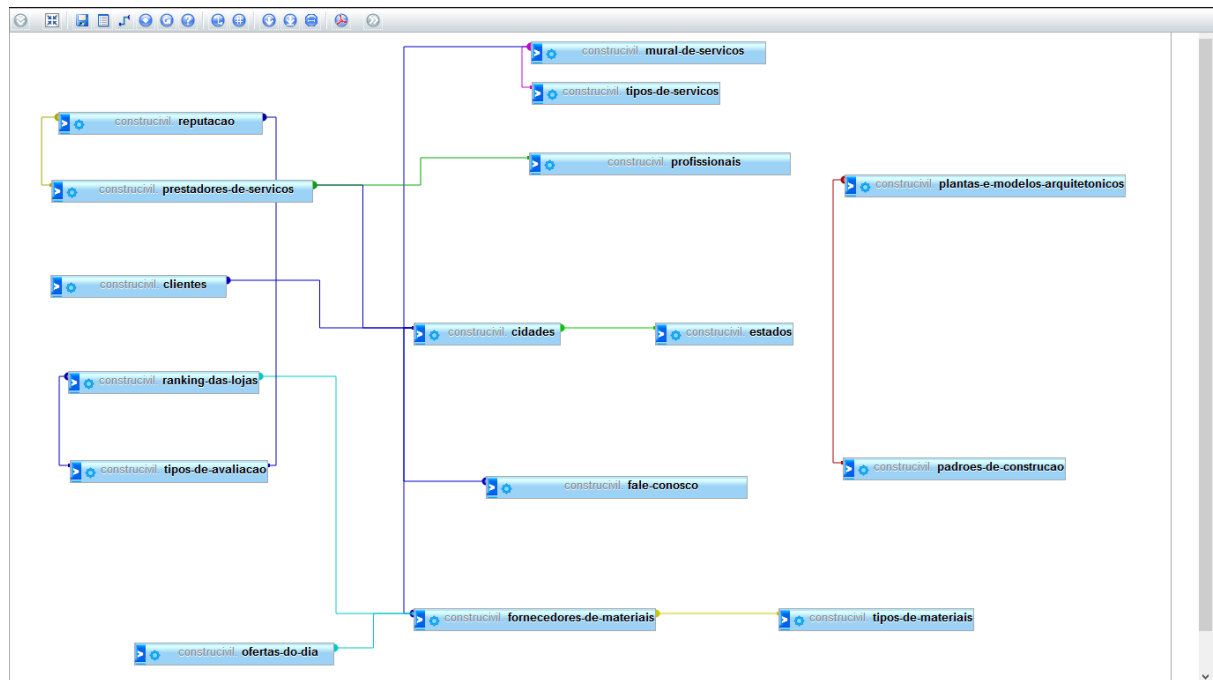


FIGURA 15: DIAGRAMA DE BANCO DE DADOS, GERADO PELA FERRAMENTA WEB PHPMYADMIN.

FONTE: O Autor (2015).

CONSIDERAÇÕES FINAIS

O banco de dados “ConstruCivil” foi desenvolvido a partir de levantamentos de requisitos e estudos de casos de uso no portal “Construtor Civil” na Internet. Foi projetado para fornecer o máximo de praticidade ao usuário final com o essencial de informações, somente o necessário, para evitar eventuais repetições de dados e, em consequência, as indesejadas redundâncias, que são prejudiciais ao funcionamento do DB.

A estrutura de banco de dados proposta neste trabalho adequou-se à realidade e dinâmicas de negócio do projeto. E, conforme uma nova demanda surgir, o DB ganhará uma nova versão, a fim de que seja páreo a novos desafios que o negócio venha, porventura, a trazer.

Em relação aos objetivos específicos propostos, foi possível encontrar as seguintes considerações:

- Facilitar o armazenamento, cadastro, edição, consulta e exclusão de dados – com determinadas informações essenciais e que não podem ser negligenciáveis já cadastradas, como Cidades, Tipos de Serviços, Profissionais, Avaliação, Padrões de Construção, Profissionais e Tipos de Materiais, o usuário poderá cadastrar, consultar e editar suas informações com segurança e sem risco de redundância nas pesquisas, porque as informações pré-cadastradas estão padronizadas;
- Permitir um maior número de possibilidades de interação com a página *Web* por diversos meios, como computadores pessoais, *notebooks*, *smartphones* e *tablets* – porque a *Web Page* Construtor Civil foi montada com a linguagem HTML5, versão mais recente que permite a página ser exibida em formato responsivo, de acordo com o dispositivo que está sendo utilizado;
- Reduzir erros de cadastro – como dito anteriormente, informações essenciais, que foram previamente cadastradas, estão padronizadas, para que o usuário não precise redigitá-las;

- Agilizar o acesso à informação – com o banco de dados integrado à *Web Page*, haverá um grande impacto no que se trata a peregrinação das pessoas na procura por produtos e serviços na área da construção civil no Litoral Paranaense. Os clientes poderão encontrar tudo o que precisam em um só lugar. Se todas as empresas e profissionais da área colaborarem voluntariamente para o preenchimento de suas informações ao sistema *Web*, a forma como se procura os produtos e serviços irá mudar radicalmente na região, possibilitando a visão de um novo panorama na construção civil no Litoral Paranaense.

Entretanto, apesar do experimento bem-sucedido deste sistema de banco de dados, este é um banco de dados relacional, e é uma tecnologia que está prestes a ser superada. Conforme mencionado, novas tecnologias de bancos de dados estão ganhando espaço no mercado aos poucos. Será apenas uma questão de tempo para que a tecnologia NoSQL, consiga uma ascensão mais rápida e a aceitação seja mais fácil por parte das empresas, especialmente as futuras.

Em virtude de tal fato, o modelo relacional, como é o caso do banco de dados “ConstruCivil”, está cada vez mais raro no mercado, por ele ter se baseado em modelo conceitual. Ainda não é possível definir com certeza sobre a aplicabilidade do NoSQL para o DB “ConstruCivil”, não necessariamente por falta de informações pertinentes ao assunto, mas sim, porque ainda não se chegou a uma definição clara sobre o termo NoSQL, nem seu funcionamento.

A partir disso, é arriscado fazer uma migração do banco desenvolvido em MySQL para a outra plataforma, por onde rompe com a antiga tradição de relacionamento entre tabelas. Contudo, se for possível aprender com o código-fonte da Google e aplicar algo ao DB “ConstruCivil”, acredita-se como possibilidade futura, que haveria um grande ganho de performance no carregamento de resultados de busca no *site* Construtor Civil.

REFERÊNCIAS

BATTISTI, Júlio. **Modelagem de dados – Final (normalização)**. Artigo. Disponível no site: <<http://imasters.com.br/artigo/7020/banco-de-dados/modelagem-de-dados-final-normalizacao/>> acesso em 04/04/2015.

FAYAD, Mohamed. SCHMIDT, Douglas C. **Object –Oriented Application Frameworks**. Artigo. Disponível no site: <<http://www.cs.wustl.edu/~schmidt/CACM-frameworks.html>>, acesso em 20/04/2015.

FONTENELLE, Leonardo. **Quando marketing, entretenimento e arquitetura andam juntos**. Artigo. Disponível no site:<http://www.aecweb.com.br/cont/a/quando-marketing-entretenimento-e-arquitetura-andam-juntos_11039> acesso em 03/04/2015.

GUIMARÃES, Arthur. **A importância de documentar as lições aprendidas no projeto**. Artigo. Disponível no site: <<http://www.projectbuilder.com.br/blog-pb/entry/blog-gestao-de-projetos/a-importancia-de-documentar-as-lico-es-aprendidas-no-projeto>>, acesso em 15/03/2015.

HEUSER, Carlos Alberto. **Projeto de banco de dados**. 6ª edição. Porto Alegre, Bookman, 2008.

LIMA, Allyn G. A. **Padrão SQL e sua evolução**. Artigo. Campinas, UNICAMP, 2004.

MACHADO, Felipe N. R. **Banco de dados: Projeto e Implementação**. 2ª edição. Ed. Ética, São Paulo, 2008.

MARTINI, Carla C. MANFREDINI, Mercedes L. CAMATTI, Tassara B. CAMARGO, Maria E. DA CRUZ, Marcia R. **O valor da programação na produção: Gerenciamento de recursos para a eficiência na construção civil**. Artigo. Qualitas. Vol. 13. Nº 1. 2012.

MILANI, André. **MySQL: guia do programador**. Novatec, São Paulo, 2006.

PADLIPSKAS, Sálvio. **As vantagens do uso de padronização em objetos de banco de dados.** Artigo. Disponível no site: <<http://www.devmedia.com.br/as-vantagens-do-uso-de-padronizacao-em-objetos-no-banco-de-dados/3221>> acesso em 04/04/2015.

SALADAGE, Pramod J. FOWLER, Martin. **NoSQL – Essencial.** São Paulo, Novatec, 2013.

SILVA, Maurício S. **HTML 5.** São Paulo, Novatec, 2011.

Site:<<http://db-engines.com/en/ranking>> acesso em 20/04/2015.

Site:<<http://educacao.globo.com/artigo/profissoes-do-futuro-tecnologia-movimenta-mercado-de-trabalho.html>> acesso em 03/04/2015.

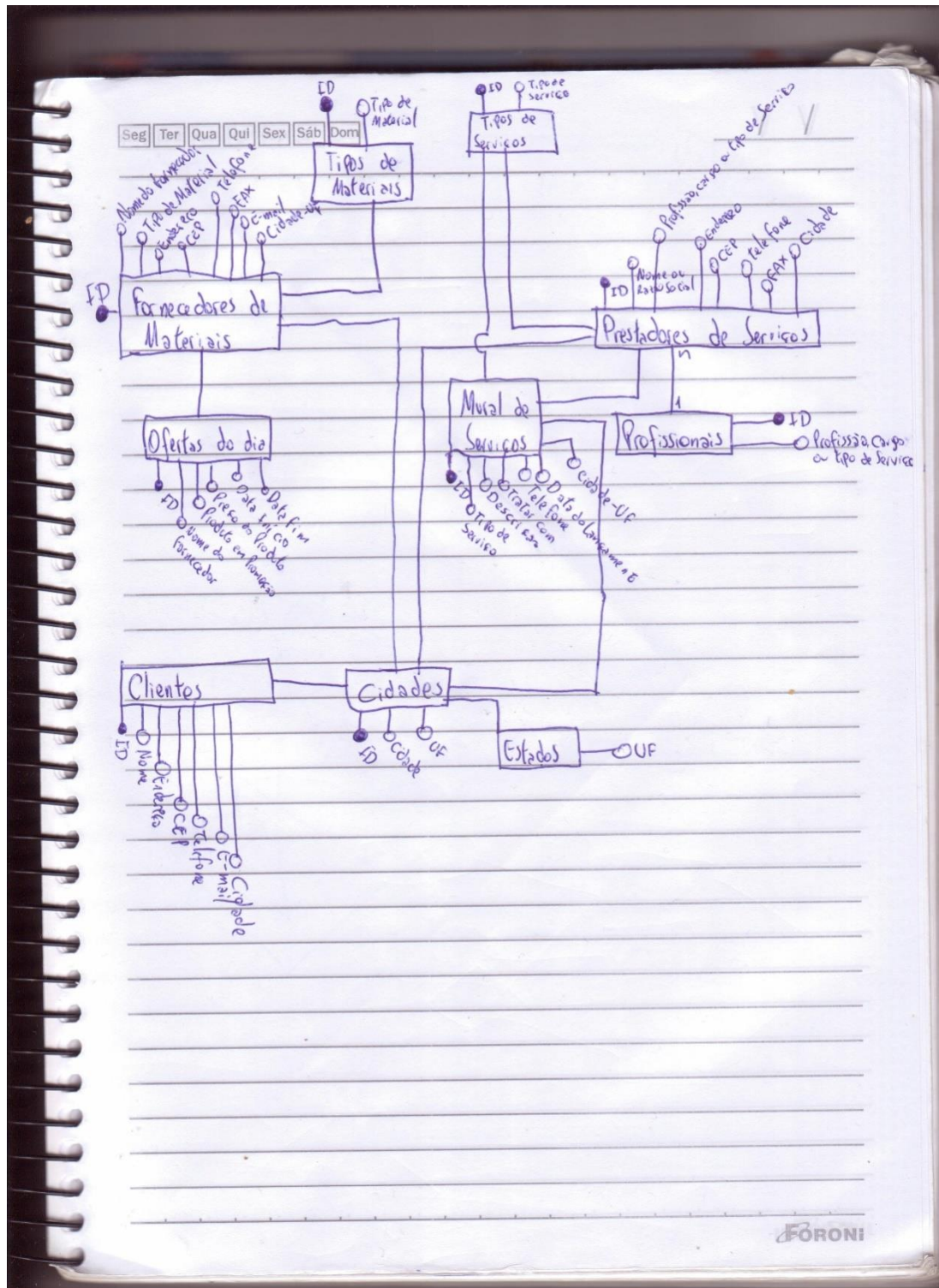
Site:<<http://portal.dataprev.gov.br/2015/03/17/unidades-de-desenvolvimento-de-software-uds/>> acesso em 20/04/2015.

Site: <<http://www.mysql.com>> acesso em 01/07/2015.

Site:<<http://portal.dataprev.gov.br/2009/10/28/previdencia-social-apresenta-siprev-para-gestores-de-rpps/>> acesso em 20/04/2015.

APÊNDICES

APÊNDICE A: PORTFÓLIO: MODELO ENTIDADE-RELACIONAMENTO DO BANCO DE DADOS "ConstruCivil"



APÊNDICE B: CÓDIGO GERADO PELO PHPMYADMIN POR MEIO DO EXPORTADOR

```
--
-- Database: `construcivil`
--

-- -----

--
create database `construcivil`;
use `construcivil`;
-- -----

--
-- Estrutura da tabela `cidades`
--

CREATE TABLE IF NOT EXISTS `cidades` (
  `Codigo-da-cidade` int(7) NOT NULL COMMENT 'Codigo da cidade segundo
cadastro do IBGE',
  `Cidade-UF` varchar(200) NOT NULL COMMENT 'Nome da cidade e a sigla da
unidade federativa a qual pertence',
  `UF` varchar(2) NOT NULL,
  PRIMARY KEY (`Codigo-da-cidade`),
  KEY `Cidade-UF` (`Cidade-UF`),
  KEY `UF` (`UF`)
) ENGINE=InnoDB DEFAULT CHARSET=latin1;

--
-- Extraindo dados da tabela `cidades`
--

INSERT INTO `cidades` (`Codigo-da-cidade`, `Cidade-UF`, `UF`) VALUES
(4118204, 'Paranagua', 'PR');

-- -----

--
-- Estrutura da tabela `clientes`
--

CREATE TABLE IF NOT EXISTS `clientes` (
```

```

`Codigo-do-cliente` int(6) NOT NULL AUTO_INCREMENT COMMENT 'Codigo
aleatorio para identificar um único cliente. Contem 6 dígitos, e usa a sintaxe Ano-
mes-nº sequencial',
`Nome-do-cliente` varchar(100) NOT NULL COMMENT 'Nome completo do cliente.
Cada cliente podera ser identificado por um único codigo, em eventuais casos de
coincidencia ou semelhanca de nomes, como Maria Aparecida da Silva ou Jose da
Silva, por exemplo.',
`Endereco` varchar(200) NOT NULL COMMENT 'Domicílio do cliente, contendo o
nome da rua, nº da casa e bairro.',
`CEP` int(8) NOT NULL COMMENT 'Codigo postal do domicílio do cliente, segundo
critério dos Correios. O número deve ser digitado sem traco.',
`E-mail` varchar(100) DEFAULT NULL COMMENT 'PREENCHIMENTO NaO-
OBRIGAToRIO. Endereco eletrônico do cliente, se tiver.',
`Telefone` varchar(20) DEFAULT NULL COMMENT 'PREENCHIMENTO NaO
OBRIGAToRIO. Aqui devera ser digitado somente números, junto com o DDD, sem
"()" nem "-".',
`Cidade-UF` varchar(200) NOT NULL COMMENT 'Neste campo, o cliente define o
local de sua residencia, em uma lista de cidades ja cadastradas no banco de dados,
na tabela "Cidades".',
PRIMARY KEY (`Codigo-do-cliente`),
KEY `clientes_ibfk_1` (`Cidade-UF`)
) ENGINE=InnoDB DEFAULT CHARSET=latin1 AUTO_INCREMENT=1 ;

```

```

-- -----

```

```

--
-- Estrutura da tabela `estados`
--

```

```

CREATE TABLE IF NOT EXISTS `estados` (
  `UF` varchar(2) NOT NULL,
  KEY `UF` (`UF`),
  KEY `UF_2` (`UF`)
) ENGINE=InnoDB DEFAULT CHARSET=latin1 COMMENT='Esta tabela vai servir à
tabela "Cidades". Relação de todos estados do Brasil.';

```

```

--
-- Extraindo dados da tabela `estados`
--

```

```

INSERT INTO `estados` (`UF`) VALUES
('AC'),
('AL'),
('AM'),
('AP'),
('BA'),

```

```

('CE'),
('DF'),
('ES'),
('GO'),
('MA'),
('MG'),
('MS'),
('MT'),
('PA'),
('PB'),
('PE'),
('PI'),
('PR'),
('RJ'),
('RN'),
('RO'),
('RR'),
('RS'),
('RS'),
('SC'),
('SP'),
('TO');

```

```

-- -----

```

```

--

```

```

-- Estrutura da tabela `fale-conosco`

```

```

--

```

```

CREATE TABLE IF NOT EXISTS `fale-conosco` (
  `Requisicao` int(10) NOT NULL AUTO_INCREMENT,
  `Nome` varchar(100) NOT NULL,
  `Telefone` varchar(20) DEFAULT NULL,
  `E-mail` varchar(200) DEFAULT NULL,
  `Site-ou-blog` varchar(100) DEFAULT NULL,
  `Facebook-Google+-Twitter-outros` varchar(200) DEFAULT NULL,
  `Cidade-UF` varchar(200) NOT NULL,
  `Descricao` text NOT NULL,
  PRIMARY KEY (`Requisicao`),
  KEY `fale-conosco_ibfk_1` (`Cidade-UF`)
) ENGINE=InnoDB DEFAULT CHARSET=latin1 AUTO_INCREMENT=1 ;

```

```

-- -----

```

```

--

```

```

-- Estrutura da tabela `fornecedores-de-materiais`

```


--

```
CREATE TABLE IF NOT EXISTS `fornecedores-de-materiais` (
  `Codigo-do-Fornecedor` int(5) NOT NULL AUTO_INCREMENT COMMENT 'Codigo
gerado automaticamente, para identificar um único fornecedor, para eventuais casos
de coincidencia ou semelhanca de nomes. Contem 5 dígitos e usa a sintaxe ano-nº
sequencial.',
  `Nome-do-Fornecedor` varchar(100) NOT NULL COMMENT 'Nome-ou-razao-
social',
  `Tipo-de-material` varchar(50) NOT NULL COMMENT 'As opcoes estao
cadastradas na tabela "Tipos-de-materiais"',
  `Endereco` varchar(100) NOT NULL COMMENT 'Endereco do estabelecimento,
contendo o nome da rua, nº e bairro.',
  `CEP` int(8) NOT NULL COMMENT 'Codigo postal segundo criterio dos Correios.
Deve ser digitado somente números e sem "-".',
  `Cidade-UF` varchar(200) NOT NULL COMMENT 'As cidades serao mostradas em
uma lista, que esta vinculada à tabela "Cidades".',
  `Telefone` varchar(20) NOT NULL COMMENT 'Como toda empresa tem, ao
menos, um número de telefone, aqui devera ser digitado somente números, junto
com o DDD, sem "()" nem "-".',
  `FAX` varchar(20) DEFAULT NULL COMMENT 'PREENCHIMENTO NaO-
OBRIGAToRIO, caso nao disponha de fax.',
  `Email` varchar(100) DEFAULT NULL COMMENT 'PREENCHIMENTO NaO-
OBRIGAToRIO, caso nao tenha e-mail.',
  `Site` text COMMENT 'PREENCHIMENTO NaO-OBRIGAToRIO, caso a empresa
nao disponha de pagina na Internet.',
  `Avaliacao` varchar(10) NOT NULL,
  PRIMARY KEY (`Codigo-do-Fornecedor`),
  KEY `fornecedores-de-materiais_ibfk_1` (`Cidade-UF`),
  KEY `fornecedores-de-materiais_tipodeMaterial` (`Tipo-de-material`),
  KEY `Nome-do-Fornecedor` (`Nome-do-Fornecedor`),
  KEY `Avaliacao` (`Avaliacao`)
) ENGINE=InnoDB DEFAULT CHARSET=latin1 AUTO_INCREMENT=15 ;
```

--

-- Extraindo dados da tabela `fornecedores-de-materiais`

--

```
INSERT INTO `fornecedores-de-materiais` (`Codigo-do-Fornecedor`, `Nome-do-
Fornecedor`, `Tipo-de-material`, `Endereco`, `CEP`, `Cidade-UF`, `Telefone`, `FAX`,
`Email`, `Site`, `Avaliacao`) VALUES
(14, 'Maria', 'Madeira', 'Rua J', 83212000, 'Paranagua', '41', NULL, NULL, NULL,
'Bom');
```

-- -----

--

-- Estrutura da tabela `mural-de-servicos`

--

```
CREATE TABLE IF NOT EXISTS `mural-de-servicos` (
  `ID` int(10) NOT NULL AUTO_INCREMENT COMMENT 'Codigo aleatorio gerado
automaticamente pelo sistema, com a sintaxe ano-mes-dia-nº sequencial. Exemplo:
1404070001',
  `Tipo-de-servico` varchar(50) NOT NULL COMMENT 'Lista de Tipos-de-servicos
previamente cadastrados na tabela "Tipos-de-servicos".',
  `Descricao` text NOT NULL COMMENT 'Campo destinado para que o cliente possa
dar detalhes maiores para o empreendimento desejado.',
  `Telefone-para-contato` varchar(20) NOT NULL COMMENT 'Somente números,
com o DDD e sem "(" nem "-". O cliente deve dispôr de um número de telefone,
para que o profissional entre em contato.',
  `Tratar-com` varchar(50) NOT NULL COMMENT 'Neste campo, o cliente pode
identificar a si mesmo ou ao titular do telefone para entrar em contato.',
  `Data-do-lancamento-da-oferta` timestamp NOT NULL DEFAULT
CURRENT_TIMESTAMP ON UPDATE CURRENT_TIMESTAMP COMMENT 'Este
campo e preenchido automaticamente, no ato do cadastro.',
  `Cidade-do-projeto` varchar(200) NOT NULL COMMENT 'Lista de cidades
previamente cadastradas, referenciadas na tabela "Cidades".',
  PRIMARY KEY (`ID`),
  KEY `Mural-de-servicos_ibfk_1` (`Cidade-do-projeto`),
  KEY `Tipo-de-servico_ibfk_Mural-de-Servicos` (`Tipo-de-servico`)
) ENGINE=InnoDB DEFAULT CHARSET=latin1 AUTO_INCREMENT=1 ;
```

-- -----

--

-- Estrutura da tabela `ofertas-do-dia`

--

```
CREATE TABLE IF NOT EXISTS `ofertas-do-dia` (
  `Codigo-da-oferta` int(10) NOT NULL AUTO_INCREMENT COMMENT 'Codigo
gerado automaticamente, contendo 9 dígitos. A sintaxe e ano-mes-dia-nº
sequencial.',
  `Nome-da-loja` varchar(100) NOT NULL COMMENT 'Nome-ou-razao-social.',
  `Produto-em-promocao` varchar(50) NOT NULL COMMENT 'A loja digita no nome
do produto na promocao. Disponível ate 50 caracteres.',
  `Preco-do-produto` double(6,2) NOT NULL COMMENT 'Separe as casas decimais
com o ponto (.) e nao com vírgula (,). Do contrario, o valor a ser informado nao
correspondera conforme o desejado.',
  `Início-da-promocao` datetime NOT NULL COMMENT 'Data completa. A sintaxe e
ano-mes-dia, como em "2014-04-07" ou "2013-05-27".',
```

```
`Termino-da-promocao` datetime NOT NULL COMMENT 'Data completa. A sintaxe
e ano-mes-dia, como em "2014-04-07" ou "2013-05-27".',
PRIMARY KEY (`Codigo-da-oferta`),
KEY `Ofertas-do-dia_ibfk_1` (`Nome-da-loja`)
) ENGINE=InnoDB DEFAULT CHARSET=latin1 AUTO_INCREMENT=1 ;
```

```
-- -----
```

```
--
```

```
-- Estrutura da tabela `padroes-de-construcao`
```

```
--
```

```
CREATE TABLE IF NOT EXISTS `padroes-de-construcao` (
  `Codigo-PC` int(11) NOT NULL COMMENT 'Codigo do Padrao-de-construcao',
  `Padrao-de-construcao` varchar(50) NOT NULL COMMENT 'O tipo padrao para
construcoes.',
  PRIMARY KEY (`Codigo-PC`),
  KEY `Padrao-de-construcao` (`Padrao-de-construcao`)
) ENGINE=InnoDB DEFAULT CHARSET=latin1;
```

```
--
```

```
-- Extraindo dados da tabela `padroes-de-construcao`
```

```
--
```

```
INSERT INTO `padroes-de-construcao` (`Codigo-PC`, `Padrao-de-construcao`)
VALUES
(14, 'Alto Padrao'),
(13, 'Comercial'),
(11, 'Ecologico'),
(12, 'Economico');
```

```
-- -----
```

```
--
```

```
-- Estrutura da tabela `plantas-e-modelos-arquiteticos`
```

```
--
```

```
CREATE TABLE IF NOT EXISTS `plantas-e-modelos-arquiteticos` (
  `ID` int(10) NOT NULL AUTO_INCREMENT COMMENT 'Codigo gerado
automaticamente, contem 7 dígitos e usa a sintaxe ano-mes-nº sequencial.',
  `Nome-do-projeto` varchar(100) NOT NULL COMMENT 'Identificacao do projeto
cadastrado pelo arquiteto.',
  `Padrao-de-construcao` varchar(50) NOT NULL COMMENT 'Aqui, o arquiteto
define qual tipo Padrao-de-construcao para o seu empreendimento.',
```

```

`Descricao` text NOT NULL COMMENT 'Aqui, o arquiteto responsavel pelo
empreendimento em questao digita todos os detalhes, de modo a fornecer ao cliente
o melhor panorama que pode convencer o cliente a adquirir o produto.',
`Data-do-projeto` timestamp NOT NULL DEFAULT CURRENT_TIMESTAMP ON
UPDATE CURRENT_TIMESTAMP COMMENT 'A data e gerada automaticamente
no ato do cadastro.',
`link` text NOT NULL COMMENT 'Aqui, o arquiteto deve inserir o endereco
eletrônico do que pode ser uma figura ilustrativa ou um arquivo de texto em PDF,
para fins de ilustracao do empreendimento.',
PRIMARY KEY (`ID`),
KEY `Plantas-e-modelos-arquitetonicos_ibfk_1` (`Padrao-de-construcao`)
) ENGINE=InnoDB DEFAULT CHARSET=latin1 AUTO_INCREMENT=1 ;

```

```

-----

```

```

--
-- Estrutura da tabela `prestadores-de-servicos`
--

```

```

CREATE TABLE IF NOT EXISTS `prestadores-de-servicos` (
  `Codigo-do-prestador-de-servico` int(5) NOT NULL AUTO_INCREMENT
  COMMENT 'Codigo gerado automaticamente. Contem 5 dígitos.',
  `Nome-ou-razao-social` varchar(100) NOT NULL COMMENT 'Nome do profissional
ou do estabelecimento.',
  `Profissao-cargo-ou-Tipo-de-servico` varchar(50) NOT NULL COMMENT 'Lista de
profissoes, cargos ou tipos de servico, referenciados na tabela "Profissionais".',
  `Endereco` varchar(100) NOT NULL COMMENT 'Endereco completo com nome da
rua, nº e bairro.',
  `CEP` int(8) NOT NULL COMMENT 'Codigo postal, segundo criterio dos Correios.
Deve ser digitado somente números e sem "-".',
  `Cidade-UF` varchar(200) NOT NULL COMMENT 'Lista referenciada na tabela
"Cidades".',
  `Telefone` varchar(20) NOT NULL COMMENT 'Numero de telefone com DDD e
sem "(" nem "-". Somente números.',
  `Fax` varchar(20) DEFAULT NULL COMMENT 'PREENCHIMENTO NaO-
OBRIGAToRIO, caso nao disponha de fax.',
  `E-mail` text COMMENT 'PREENCHIMENTO NaO-OBRIGAToRIO, caso nao
disponha de correio eletrónico.',
  `Site` text COMMENT 'PREENCHIMENTO NaO-OBRIGAToRIO, caso nao
disponha de pagina na Internet.',
  PRIMARY KEY (`Codigo-do-prestador-de-servico`),
  KEY `Prestadores-de-servicos_ibfk_1` (`Cidade-UF`),
  KEY `Prestadores-de-servicos_ibfk_Profissionais` (`Profissao-cargo-ou-Tipo-de-
servico`),
  KEY `Nome-ou-razao-social` (`Nome-ou-razao-social`)
) ENGINE=InnoDB DEFAULT CHARSET=latin1 AUTO_INCREMENT=1 ;

```

```
-- -----
```

```
--
```

```
-- Estrutura da tabela `profissionais`
```

```
--
```

```
CREATE TABLE IF NOT EXISTS `profissionais` (
  `Profissao-cargo-ou-Tipo-de-servico` varchar(50) NOT NULL COMMENT 'Aqui esta
uma lista de ofícios relacionados à area de construcao civil.',
  PRIMARY KEY (`Profissao-cargo-ou-Tipo-de-servico`)
) ENGINE=InnoDB DEFAULT CHARSET=latin1;
```

```
--
```

```
-- Extraindo dados da tabela `profissionais`
```

```
--
```

```
INSERT INTO `profissionais` (`Profissao-cargo-ou-Tipo-de-servico`) VALUES
('Arquiteto'),
('Azulejista'),
('Carpinteiro'),
('Decorador'),
('Eletricista'),
('Encanador'),
('Engenheiro Civil'),
('Engenheiro Eletricista'),
('INFORMADO PELO CLIENTE'),
('Jardineiro'),
('Marceneiro'),
('Mestre de obras'),
('Paisagista'),
('Pedreiro'),
('Pintor'),
('Projetista'),
('Rocadeiro'),
('Serralheiro'),
('Vidraceiro');
```

```
-- -----
```

```
--
```

```
-- Estrutura da tabela `ranking-das-lojas`
```

```
--
```

```
CREATE TABLE IF NOT EXISTS `ranking-das-lojas` (
```

```

`Loja` varchar(100) NOT NULL COMMENT 'Lista de estabelecimentos,
referenciada na tabela "Fornecedores-de-materiais".',
`Avaliacao` varchar(10) NOT NULL COMMENT 'Lista de possíveis avaliacoes,
referenciada na tabela "Tipos-de-avaliacao".',
KEY `Loja` (`Loja`),
KEY `Avaliacao` (`Avaliacao`)
) ENGINE=InnoDB DEFAULT CHARSET=latin1;

```

```

--
-- Extraindo dados da tabela `ranking-das-lojas`
--

```

```

INSERT INTO `ranking-das-lojas` (`Loja`, `Avaliacao`) VALUES
('Maria', 'Bom');

```

```

-- -----

```

```

--
-- Estrutura da tabela `reputacao`
--

```

```

CREATE TABLE IF NOT EXISTS `reputacao` (
  `Nome-ou-razao-social` varchar(100) NOT NULL COMMENT 'Lista de profissionais
liberais ou autônomos cadastrados no sistema. Este campo está vinculado à tabela
"Prestadores-de-servicos".',
  `Avaliacao` varchar(10) NOT NULL COMMENT 'Lista de possíveis avaliacoes,
referenciada na tabela "Tipos-de-avaliacao".',
  KEY `Nome-ou-razao-social` (`Nome-ou-razao-social`),
  KEY `Avaliacao` (`Avaliacao`)
) ENGINE=InnoDB DEFAULT CHARSET=latin1;

```

```

-- -----

```

```

--
-- Estrutura da tabela `tipos-de-avaliacao`
--

```

```

CREATE TABLE IF NOT EXISTS `tipos-de-avaliacao` (
  `Avaliacao` varchar(10) NOT NULL COMMENT 'Nível de satisfacao dos usuarios.',
  KEY `Avaliacao` (`Avaliacao`),
  KEY `Avaliacao_2` (`Avaliacao`)
) ENGINE=InnoDB DEFAULT CHARSET=latin1;

```

```

--
-- Extraindo dados da tabela `tipos-de-avaliacao`
--

```

```
INSERT INTO `tipos-de-avaliacao` (`Avaliacao`) VALUES
('Bom'),
('Excelente'),
('otimo');
```

```
-- -----
```

```
--
```

```
-- Estrutura da tabela `tipos-de-materiais`
```

```
--
```

```
CREATE TABLE IF NOT EXISTS `tipos-de-materiais` (
  `Tipo-de-material` varchar(50) NOT NULL COMMENT 'Especializacao a qual um
profissional ou um estabelecimento esta focado.',
  PRIMARY KEY (`Tipo-de-material`)
) ENGINE=InnoDB DEFAULT CHARSET=latin1;
```

```
--
```

```
-- Extraindo dados da tabela `tipos-de-materiais`
```

```
--
```

```
INSERT INTO `tipos-de-materiais` (`Tipo-de-material`) VALUES
('Esquadrias metalicas'),
('Madeira'),
('Marmoraria'),
('Materiais de Construcao'),
('Materiais Eletricos'),
('Materiais hidraulicos'),
('Outros Materiais'),
('Telhas'),
('Tintas');
```

```
-- -----
```

```
--
```

```
-- Estrutura da tabela `tipos-de-servicos`
```

```
--
```

```
CREATE TABLE IF NOT EXISTS `tipos-de-servicos` (
  `Tipo-de-servico` varchar(50) NOT NULL COMMENT 'Nome de um servico
especializado.',
  PRIMARY KEY (`Tipo-de-servico`)
) ENGINE=InnoDB DEFAULT CHARSET=latin1;
```

```
--
```

```
-- Extrair dados da tabela `tipos-de-servicos`
--
```

```
INSERT INTO `tipos-de-servicos` (`Tipo-de-servico`) VALUES
('Arquitetura'),
('Construcao'),
('Construtores Especializados'),
('Decoracao'),
('Encanamento'),
('Engenharia'),
('Instalacoes Eletricas'),
('Outros'),
('Paisagismo'),
('Reparos'),
('Terraplanagem');
```

```
--
-- Constraints for dumped tables
--
```

```
--
-- Limitadores para a tabela `cidades`
--
```

```
ALTER TABLE `cidades`
  ADD CONSTRAINT `cidades_ibfk_1` FOREIGN KEY (`UF`) REFERENCES
`estados` (`UF`);
```

```
--
-- Limitadores para a tabela `clientes`
--
```

```
ALTER TABLE `clientes`
  ADD CONSTRAINT `clientes_ibfk_1` FOREIGN KEY (`Cidade-UF`) REFERENCES
`cidades` (`Cidade-UF`);
```

```
--
-- Limitadores para a tabela `fale-conosco`
--
```

```
ALTER TABLE `fale-conosco`
  ADD CONSTRAINT `fale-conosco_ibfk_1` FOREIGN KEY (`Cidade-UF`)
REFERENCES `cidades` (`Cidade-UF`);
```

```
--
-- Limitadores para a tabela `fornecedores-de-materiais`
--
ALTER TABLE `fornecedores-de-materiais`
```



```

ADD CONSTRAINT `avaliacao_ibfk_1` FOREIGN KEY (`Avaliacao`)
REFERENCES `tipos-de-avaliacao` (`Avaliacao`),
ADD CONSTRAINT `fornecedores-de-materiais_ibfk_1` FOREIGN KEY (`Cidade-
UF`) REFERENCES `cidades` (`Cidade-UF`),
ADD CONSTRAINT `fornecedores-de-materiais_tipodeMaterial` FOREIGN KEY
(`Tipo-de-material`) REFERENCES `tipos-de-materiais` (`Tipo-de-material`);

--
-- Limitadores para a tabela `mural-de-servicos`
--
ALTER TABLE `mural-de-servicos`
ADD CONSTRAINT `Mural-de-servicos_ibfk_1` FOREIGN KEY (`Cidade-do-
projeto`) REFERENCES `cidades` (`Cidade-UF`),
ADD CONSTRAINT `Tipo-de-servico_ibfk_Mural-de-Servicos` FOREIGN KEY
(`Tipo-de-servico`) REFERENCES `tipos-de-servicos` (`Tipo-de-servico`);

--
-- Limitadores para a tabela `ofertas-do-dia`
--
ALTER TABLE `ofertas-do-dia`
ADD CONSTRAINT `Ofertas-do-dia_ibfk_1` FOREIGN KEY (`Nome-da-loja`)
REFERENCES `fornecedores-de-materiais` (`Nome-do-Fornecedor`);

--
-- Limitadores para a tabela `plantas-e-modelos-arquitetonicos`
--
ALTER TABLE `plantas-e-modelos-arquitetonicos`
ADD CONSTRAINT `Plantas-e-modelos-arquitetonicos_ibfk_1` FOREIGN KEY
(`Padrao-de-construcao`) REFERENCES `padroes-de-construcao` (`Padrao-de-
construcao`);

--
-- Limitadores para a tabela `prestadores-de-servicos`
--
ALTER TABLE `prestadores-de-servicos`
ADD CONSTRAINT `Prestadores-de-servicos_ibfk_1` FOREIGN KEY (`Cidade-
UF`) REFERENCES `cidades` (`Cidade-UF`),
ADD CONSTRAINT `Prestadores-de-servicos_ibfk_Profissionais` FOREIGN KEY
(`Profissao-cargo-ou-Tipo-de-servico`) REFERENCES `profissionais` (`Profissao-
cargo-ou-Tipo-de-servico`);

--
-- Limitadores para a tabela `ranking-das-lojas`
--
ALTER TABLE `ranking-das-lojas`

```

```
ADD CONSTRAINT `RankingLoja_ibfk_1` FOREIGN KEY (`Loja`) REFERENCES
`fornecedores-de-materiais` (`Nome-do-Fornecedor`),
ADD CONSTRAINT `RankingLoja_ibfk_2` FOREIGN KEY (`Avaliacao`)
REFERENCES `tipos-de-avaliacao` (`Avaliacao`);
```

```
--
```

```
-- Limitadores para a tabela `reputacao`
```

```
--
```

```
ALTER TABLE `reputacao`
ADD CONSTRAINT `reputacao_ibfk_1` FOREIGN KEY (`Nome-ou-razao-social`)
REFERENCES `prestadores-de-servicos` (`Nome-ou-razao-social`),
ADD CONSTRAINT `reputacao_ibfk_3` FOREIGN KEY (`Avaliacao`)
REFERENCES `tipos-de-avaliacao` (`Avaliacao`);
```

```
/*!40101 SET CHARACTER_SET_CLIENT=@OLD_CHARACTER_SET_CLIENT */;
/*!40101 SET
CHARACTER_SET_RESULTS=@OLD_CHARACTER_SET_RESULTS */;
/*!40101 SET COLLATION_CONNECTION=@OLD_COLLATION_CONNECTION */;
```